PREDICTIVE ANALYTICS FOR COMPLEX ENGINEERING SYSTEMS USING HIGH-DIMENSIONAL SIGNALS

A Dissertation
Presented to
The Academic Faculty

By

Xiaolei Fang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology

May 2018

PREDICTIVE ANALYTICS FOR COMPLEX ENGINEERING SYSTEMS USING HIGH-DIMENSIONAL SIGNALS

Approved by:

Dr. Nagi Gebraeel, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Kamran Paynabar, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Jianjun (Jan) Shi
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Edmond Chow
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Tuo Zhao
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: April 2, 2018

To my beloved parents, wife Weiping, and daughter Emily.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisors Professors Nagi Gebraeel and Kamran Paynabar for their guidance, support, and encouragement throughout my doctoral study. I have been very lucky and honored to be supervised by them and have learned tremendously from them. The invaluable experience working with them will benefit me throughout my entire life.

I am also extremely grateful to Professor Jianjun Shi, who not only introduced me to my advisors and encouraged me to pursue my doctoral degree at the Georgia Institute of Technology but also has been continuously helping and encouraging me for many years. My gratitude also goes to Professor Edmond Chow and Professor Tuo Zhao for serving on my thesis committee and for their constructive comments and insightful advice to improve this dissertation.

I would also like to thank Professor Paul Kvam and Professor Alan Erera for giving me the continued opportunity and support to succeed in their graduate program. Moreover, I am thankful to all the other faculty and staff members in the H. Milton Stewart School of Industrial and Systems Engineering who have taught me and helped me throughout my graduate studies.

Special thanks go to Professor Jinwu Xu, Professor Debin Yang, Professor Jianhong Yang, Professor Min Li, and Professor Fei He, for their guidance when I was a graduate student at the University of Science and Technology Beijing, and for their support in my decision to start my doctoral study at the Georgia Institute of Technology.

My Ph.D. study would be less colorful and less exciting without my fellow students and friends. I am very thankful to the colleagues in my advisors' research groups: Dr. Linkan Bian, Dr. Rensheng Zhou, Dr. Murat Yildirim, Dr. Hao Yan, Dr. Li Hao, Dr. Chitta Ranjan, Dr. Tangbin Xia, Mr. Mostafa Reisi, Mr. Benjamin Peters, Mr. Liexiao Ding, Ms. Dan Li, Mr. Naipeng Li, Ms. Samaneh Ebrahimi, Ms. Ana Maria Estrada Gomez, Ms. Jiachen

Shi, Mr. Paritosh Ramanan, Ms. Beste Basciftci, and Dr. Salah Haridy. In addition, I would also like to express my gratefulness to my friends in ISyE, including, but not limited to, Dr. Xiaowei Yue, Dr. Weijun Xie, Dr. Rui Gao, Dr. Can Zhang, Dr. Cheng Zhou, Dr. Xingchang Wang, Dr. Yu Zhang, Dr. Xiaole Han, Mr. Tony Yaacoub, Mr. Geet Lahoti, Ms. Yasaman Mohammad Shahi, Mr. Yuchen Wen, Mr. Andi Wang, Mr. Mohammed Nabhan, Ms. Xinran Shi, Mr. Ruizhi Zhang, Ms. Juan Du, and Mr. Zhen Zhong.

Last but not least, I would like to thank my parents and my sister for their endless support. I also owe tremendously to my wife Weiping, who has given me unconditional and unlimited love and support. The love and support from my family always give me the courage and strength to overcome any difficulties in my life. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis presents new predictive analytics methodologies that extract information from massive and complex-structured high-dimensional signals with the goal of predicting (in real-time) the future state-of-health of complex engineering systems. Chapter 1 discusses the research background, motivation, and challenges, and briefly introduces the research topics in this dissertation.

Chapter 2 develops a three-step multi-sensor prognostics methodology that utilizes multistream signals to predict residual useful lifetimes of partially degraded systems. The methodology first identifies the informative sensors via the penalized (log)-location-scale regression. Then, we fuse the degradation signals of the informative sensors using multivariate functional principal component analysis (FPCA), which is capable of modeling the cross-correlation of signal streams. Finally, the third step focuses on utilizing the fused signal features for prognostics via adaptive penalized (log)-location-scale regression.

Chapter 3 presents a scalable prognostics model for applications with large scale datasets. The proposed method first develops two multistream signal fusion algorithms, multivariate FPCA and hierarchical FPCA, to effectively fuse the degradation signals from various sensors. Next, the fused features are used to dynamically predict of remaining lifetimes via an adaptive functional (log)-location scale regression model. In order to address the computational challenge, the proposed model incorporates a randomized low-rank approximation algorithm, which can help to speed up matrix decomposition for multivariate FPCA and hierarchical FPCA but without affecting the prediction accuracy.

Chapter 4 develops a semi-parametric approach that can use incomplete degradation signals from a single sensor to predict the remaining lifetime of partially degraded systems. First, key signal features are identified by applying FPCA to the available historical data, and an algorithm developed from Bayesian linear regression is used to estimate signal features using incomplete signal observations. Next, an adaptive functional regression

model is used to model the extracted signal features and the corresponding times-to-failure. The model is then used to predict remaining lifetimes and to update these predictions using real-time signals observed from fielded components.

Chapter 5 presents a robust prognostics model for multi-sensor applications with missing data. The model first fuses multistream signals by utilizing multivariate FPCA. To estimate the fused features from incomplete signals, two computationally efficient algorithms are developed. Next, a prognostics model is built by regressing the fused features against times-to-failure via (log)-location-scale regression.

Chapter 6 develops a novel supervised dimension reduction-based prognostic methodology that works for applications with multi-stream incomplete signals and censored historical failure times. The method builds an optimization problem that combines a feature extraction term and a regression term. The feature extraction term is capable of extracting low-dimensional features from multi-stream incomplete degradation signals, and the regression term regresses the features against censored failure times. By simultaneously optimizing the two terms, the censored failure times are used to supervise the feature extraction process, and thus the extracted features are guaranteed to be most informative for failure time prediction. To solve the optimization problem, we develop a Block Prox-Linear Coordinate Descent algorithm and theoretically prove its global convergence property.

Chapter 7 proposes a methodology for residual useful lifetime prediction of a system using a sequence of degradation images. The methodology integrates tensor linear algebra with traditional location-scale regression widely used in reliability and prognostics. To address the high dimensionality challenge, the coefficient tensor is decomposed using CANDECOMP/PARAFAC and Tucker decompositions, which enables parameter estimation in a high-dimensional setting. Two optimization algorithms with a global convergence property are developed for model estimation.

Chapter 8 concludes this dissertation and outlines some topics for future research.

CHAPTER 1
INTRODUCTION

1.1 Background and motivation

Industrial predictive analytics has multiple facets that range from failure predictability and optimal asset management to high-level operational and managerial insights. Failure predictability and asset management can have far-reaching implications ranging from significant economic losses to endangering human life. As a result, the health condition of modern industrial assets are often monitored by using sensing technologies through a process known as condition monitoring. In cases where the sensor signals possess trends that are strongly correlated with the progression of physical *degradation*–an irreversible damage accumulation process that results in the failure of engineering systems–they can be very useful for prognostics, which focuses on using degradation-based signals to predict time-to-failure (TTF) and operational risk of assets.

Most of the existing prognostics methodologies focus on single-sensor applications. Examples include using random coefficients models [1, 2], Brownian motion process [3, 4, 5], Gamma process [6, 7, 8], and Markov chains [9, 10]. However, many capital-intensive assets used in the energy, manufacturing, and service sectors (such as gas turbines, boilers, paper mills, and steel mills) are equipped with numerous sensors to monitor their physical performance and the operational characteristics. For example, a typical gas turbine is equipped with over 2,000 sensors that are used to monitor vibrations, temperatures, and pressures related to its health condition and a nuclear power plant typically consists of tens of thousands of variables to monitor the performance of many of its components. These sensors generate large amounts of data, which usually comes in various forms: (a) multivariate time series, (b) profile data where a single data observation represents thousands

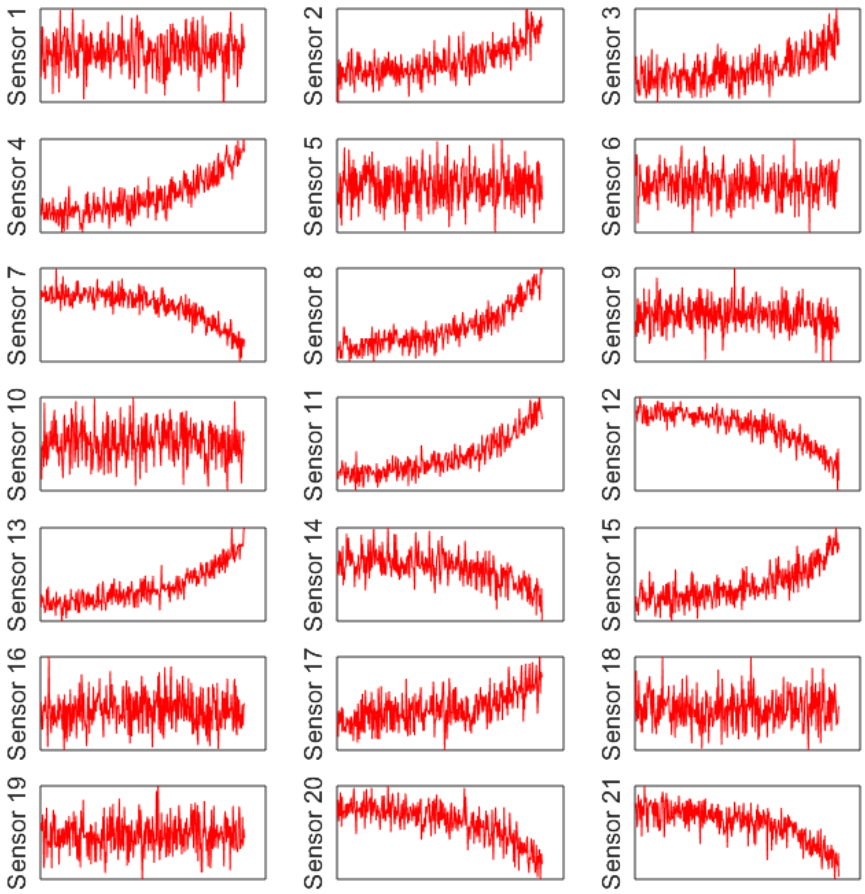


Figure 1.1: Multi-stream degradation signals of an aircraft turbofan engine from NASA data repository.

of data points, and (c) image data. For example, Figure 1.1 illustrates multivariate time series-based degradation signals of a commercial aircraft turbofan engine from NASA data repository [11]. Each engine is monitored by 21 sensors and each sensor generates a time series signal. Figure 1.2 is an example of profile-based degradation signals. The data comes from a electric power generation and storage (EPGS) systems that provides power to crank the engine during vehicle starting. Each time the engine starts, the current of the starter motor in the EPGS system is monitored and a current profile is recorded. If the current profile gets recorded throughout the life cycle of the starter motor, then a profile stream is generated. The profile stream contains the degradation information of the starter motor and thus can be modeled for prognostics purposes. Figure 1.3 shows an infrared image-based degradation signal from a rotating machinery [12]. The machinery is designed to

run degradation experiments for thrust rolling element bearings. Using the machinery, a bearing can be run from brand new to failure. During the experiment, an infrared camera is used to monitor the degradation process of the bearing and infrared images are recorded over time, which results in a degradation image stream. In this thesis, we collectively refer to the aforementioned three types of data as *high-dimensional signals*. The overarching objective of this thesis is to develop prognostic approaches that use high-dimensional signals to predict the TTF (or residual useful lifetime, RUL) of complex engineering systems that are operating in the field.

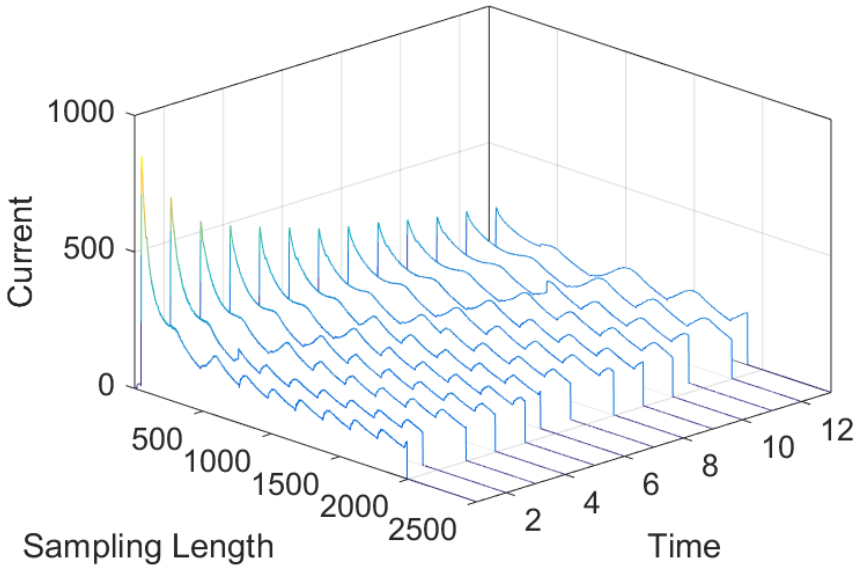


Figure 1.2: Current-based degradation profiles of a car engine starter motor.

1.2 Data characteristics and challenges

Prognostics using high-dimensional signals is an important yet challenging research topic. In this section, we discuss the common characteristics and modeling challenges of high-dimensional degradation signals.

(1) High dimensionality and high volume. Complex industrial assets are often monitored by numerous sensors, which usually generate high-dimensional signals. For multi-sensor applications, the simplest case is that a sensor produces one observation at each

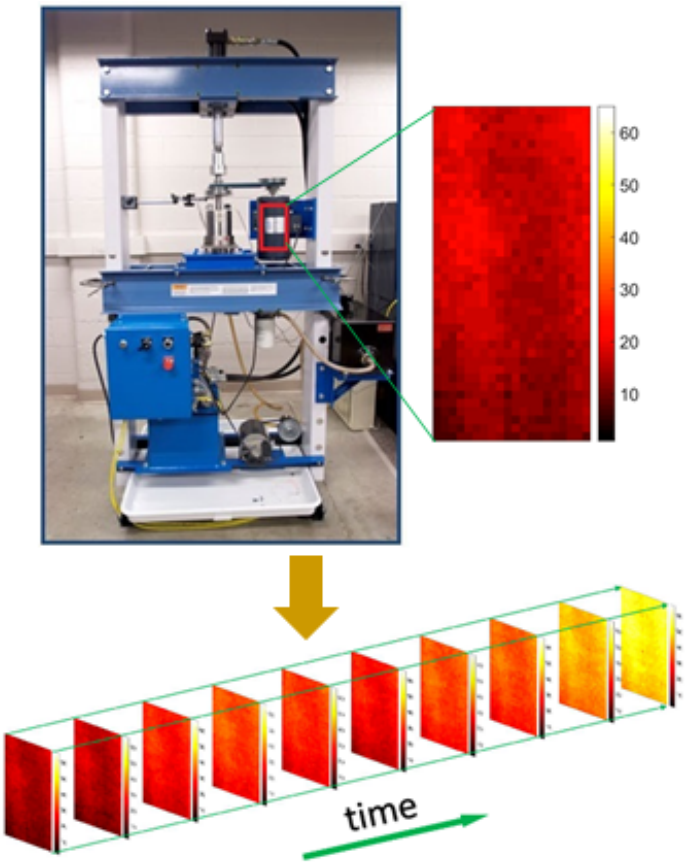


Figure 1.3: An infrared image-based degradation signal from a rotating machinery.

sampling time, which results in a time series signal with hundreds or thousands of observations over time. In some complicated cases, a sensor samples a profile at each time point, and the resulting data is a profile stream. For these cases that image sensing techniques are used, the dimensionality of data generated is even higher–in addition to the large number of pixels in each image, the number of images linearly grows over time as new images are recorded. The high dimensionality of each sensor signal coupled with the large number of sensors produce prohibitive volume of data , which raises significant **scalability** challenges for prognostic models.

(2) High velocity. With the development of sensing technology, many sensing techniques generate data at a fast rate. For example, a typical vibration sensor samples data at a rate that is easily higher than 10,000 Hz [13], and a high-speed industrial camera usually scan a product surface with the rate of more than 80 million pixels per second [14]. Such

high data collection rate poses significant **computational** challenges for *real-time* predictive analytics and requires that the prognostic models to be computationally high efficient.

(3) Complex correlation structure. High-dimensional degradation signals always exhibit complex correlation structures. One of such complex structures is auto- and cross-correlation. For example, in multi-sensor applications, the degradation signal from each sensor is auto-correlated over time, and degradation signals from different sensors are cross-correlated since they capture different aspects of the same degradation process. Another common complex structure is spatio-temproal correlation. For instance, in the degradation image stream in Figure 1.3, pixels within an image are spatially correlated, and corresponding pixels across sequential images are temporally correlated. The complex correlation structures pose significant **analytical** challenges. In order to better characterize the underlying degradation process and achieve more accurate prediction capability, the complex correlation pattern should be modeled carefully.

(4) Poor data quality. Most industrial predictive analytics models are developed based on the premise that sensor data is observed and collected continuously with no interruptions. In reality, however, industrial assets operate in harsh environments that generate errors in data acquisition, communication, read/write operations, etc, and thus the resulting degradation observations often contain outliers as well as missing and corrupt data. Such poor data quality settings pose enormous challenges for predictive analytics modeling and require prognostics models to be **robust** to data quality.

This dissertation focuses on developing new predictive analytics methodologies that address the aforementioned scalability, computational, analytical, and robustness challenges of analyzing high-dimensional degradation signals with the goal of predicting (in real-time) the future state-of-health of complex engineering systems.

1.3 Overview of dissertation

In this section, we briefly discuss the research topics in this dissertation.

1.3.1 Multistream sensor fusion-based prognostics model for systems with single failure modes

In many industrial applications, systems are equipped with multiple sensors to monitor their condition. Raw signals from these sensors are often transformed into degradation signals that can be used to predict TTF. Most of the existing literature focuses either on modeling an individual degradation signal or on combining all the degradation signals together through some sort of fusion mechanism. However, for such multi-sensor applications, there always exists some level of redundancy among the sensors. That is, more than one sensor may capture the same physical effects to a similar degree. In some other instances, signals from some sensors may have little or no relation to the underlying physics, thus compromising the accuracy of predicting failures. As a result, selecting the appropriate sensors before the fusion process may possibly lead to better failure predictability.

This research topic proposes a multi-sensor prognostic methodology that incorporates a systematic sensor selection procedure. Our methodology consists of three steps. The first step is a formalized sensor selection algorithm that systematically identifies the most informative sensors that should be used to predict failure and TTF. This step is based upon a penalized variable selection methodology [15, 16]. The goal is to identify highly informative sensors that when combined together provide a relatively comprehensive yet precise characterization of the underlying physical degradation. The second step focuses on intelligently combining the degradation signals associated with the informative sensors identified by the sensor selection process. This is achieved by developing a signal fusion algorithm based on multivariate FPCA [17]. multivariate FPCA focuses on capturing the joint variation of multistream functional data. The benefit of using multivariate FPCA is that it reduces dimensionality of the data and provides fused signal features in the form of MFPC-scores. Finally, the third step focuses on utilizing the fused signal features for prognostics. This will be accomplished by using an adaptive penalized (log)-location-scale (LLS) regression model, which estimates TTF and provides the means to continuously up-

date these estimates as real-time signals are observed from fielded systems.

1.3.2 Scalable prognostic models for large-scale condition monitoring applications

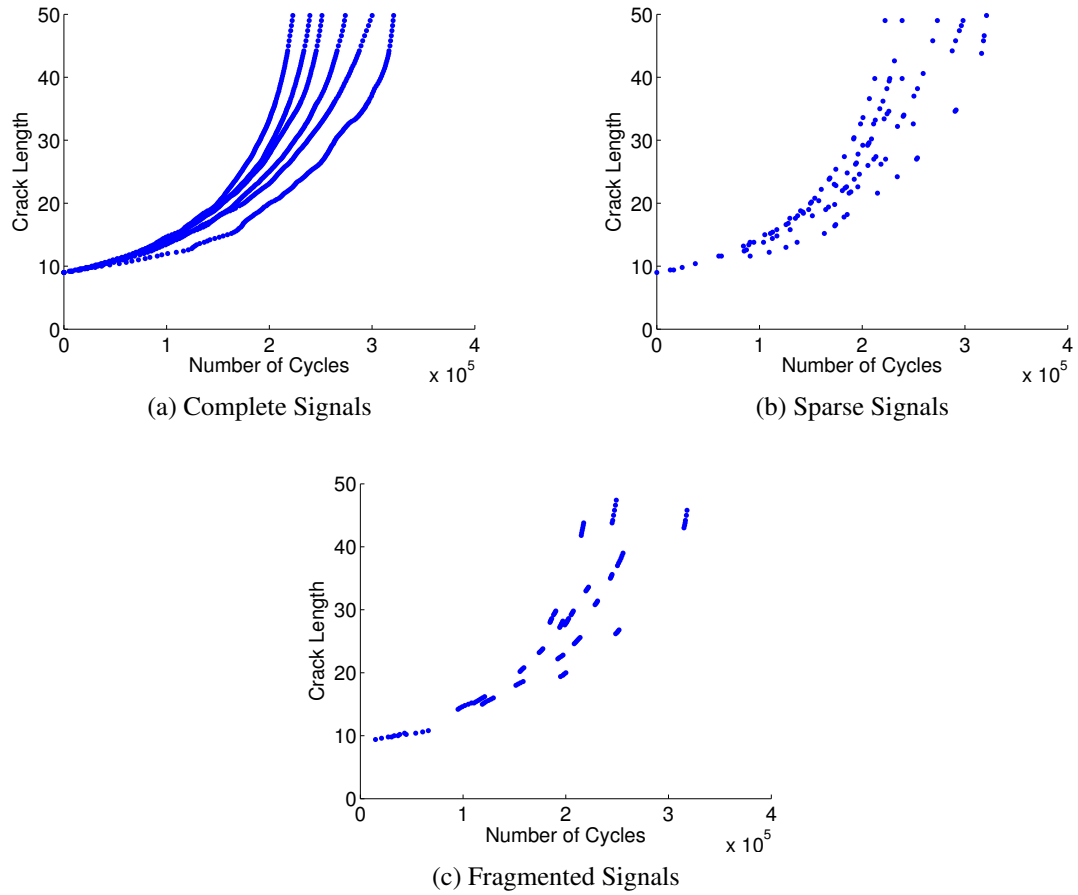
The volume and dimensionality of condition monitoring data generated by many industrial applications has become prohibitive. For example, some optical sensors used for turbine blade crack detection generate 600 gigabytes per day – almost 7 times Twitter daily volume [18]. However, the existing predictive analytic algorithms are not designed to scale with such Big Data settings. To address this challenge, this research topic develops a prognostic modeling framework that is scalable with the size of the condition monitoring data.

Our methodology is based on using tools from functional data analysis to systematically extract and combine features from different degradation signals, and subsequently use these features to predict remaining lifetimes of partially degraded equipment. Specifically, we use FPCA to develop signal fusion algorithms, and functional regression to predict the remaining lifetime. Two signal fusion algorithms, namely Multivariate FPCA and Hierarchical FPCA, are developed. Both algorithms are extensions of FPCA that allows us to utilize the FPCA framework for multi-sensor applications. Multivariate FPCA works by concatenating various types of degradation signals, thus the resulting signal matrix becomes even much larger. Hierarchical FPCA works by first applying FPCA to the individual degradation signals (grouped by sensor type), and then extracting their corresponding FPC-scores. Next, it concatenates these FPC-scores and computes a set of fused signal features by applying regular PCA on the concatenated vector. Both Multivariate FPCA and Hierarchical FPCA are able to capture the cross-correlation among signals from different sensors and provide fused features. To predict remaining lifetimes, we use adaptive functional LLS regression to model the relationship between the fused features and TTF.

FPCA is inherently computationally expensive because it involves matrix decomposition, e.g., singular value decomposition and/or eigen decomposition. In large scale settings involving large amounts of data, this aspect can become a major impediment to the scalabil-

ity of FPCA. This problem becomes even more prominent in the multi-sensor applications like the ones considered in this research topic. From another perspective, functional regression, in its classical sense, is used for one-shot estimation of the response variable, TTF in our case. Our goal, however, is to be able to integrate real-time degradation signals observed from fielded equipment to update predictions of remaining lifetime on a constant basis. To achieve this, we exploit an adaptive version of functional regression known as time-varying functional regression [19]. Time-varying functional regression allows us to recalibrate our model based on the unique degradation signals of each unit. However, this process results in a new signal matrix each time. As a result, matrix decomposition needs to be performed repeatedly as new data becomes available from the field.

We address the computational challenges by leveraging recent developments in randomized algorithms used for numerical linear algebra. Specifically, we utilize randomized low-rank approximation (RLA) in key steps within the FPCA methodology. RLA focuses on building a matrix with the smallest rank but preserves most of the useful information of the original matrix. It works by first computing an approximation to the range (also known as column space) of a matrix via randomized sampling. In our case the matrix of concern is the signal matrix. The signal matrix is then projected to the approximated range, and a factorization of the resulting low-rank matrix is computed. Although RLA is an approximation technique, its error bounds have been well-studied [20]. One of the key contributions of this research topic is that we enhance the scalability of FPCA by exploiting RLA. However, this integration is not trivial. For example, a key aspect in RLA is that is requires that the rank (number of principal components) of the matrix be known in advance, which is not the case in our framework. Details of the integration are discussed in Chapter 3.



(a) Complete Signals

(b) Sparse Signals

(c) Fragmented Signals

Figure 1.4: Examples of complete, sparse and fragmented degradation signals from crack growth data.

1.3.3 An adaptive functional regression-based prognostic model for applications with missing data

Degradation-based prognostics models focus on characterizing how degradation signals evolve over time and using degradation signals to predict and update remaining lifetime in real-time. A large number of -based prognostics models have been proposed in the literature. However, the effectiveness of these models relies primarily on the fidelity of parameter estimation, which is often driven by the characteristics and the quality of historical data. For example, most models assume that historical degradation signals are completely and, for all practical purposes, continuously observed from an "as good as new" state up to

the point of failure. In reality, however, continuous or frequent observations of degradation is not always possible nor economical. Examples of such scenarios include monitoring cracks on gas turbine blades that require shutting down the turbine or assessing the concentration of dissolved gases in transformers. Other examples may involve sensor failure or disconnection. Therefore, in practice, it is more likely that degradation signals are observed randomly or at intermittent points in time resulting in sparse or fragmented signal observations as illustrated in Figure 1.4.

If parametric models are used to model signals with such high levels of missing data, it is likely that the available data will not be enough to accurately identify a suitable trend or general path for the degradation signals. In Chapter 4, we utilize a semi-parametric approach to develop a prognostic degradation model for sparse and fragmented signals. First, FPCA is used to identify key features of the incomplete signals. FPCA provides a low-dimensional and parsimonious representation of each curve by reducing it to a set of FPC-scores. The FPC-scores estimated using signals with missing observations are likely to be similar to those that would have been estimated if all observation were present. Once the signal features are extracted, an adaptive functional regression model is used to model the relationship between the FPC-scores and historical TTFs. The proposed framework provides a means to incorporate in-situ signals observed from partially degraded components in the field in order to update the model. The updated model is then used to revise the predicted remaining lifetimes.

1.3.4 Multi-sensor prognostic modeling for applications with highly incomplete signals: A matrix completion approach

Most of the existing prognostics models for multi-sensor settings are designed for applications with *complete* degradation signals, that is, signals are observed *continuously and frequently at regular time grids*. In reality, however, degradation observations often contain outliers as well as missing and corrupt data. We refer to such phenomena as *incomplete*

degradation signals. Figure 1.5 shows examples of *complete* (i.e., the gray trajectories) and *incomplete* (i.e., the solid dots) degradation signals.

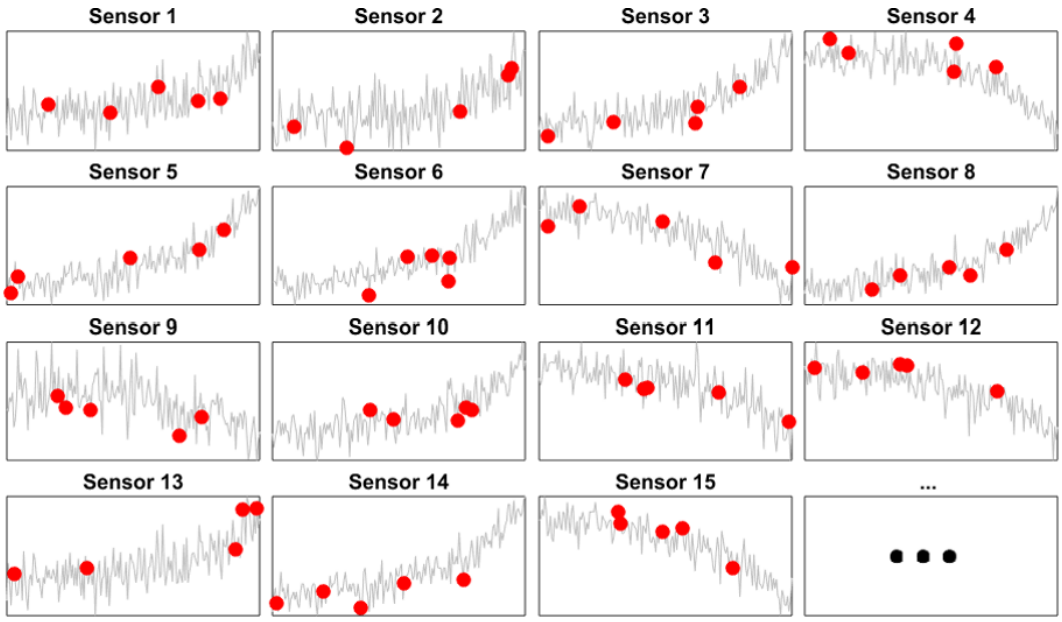


Figure 1.5: An illustration of complete and incomplete degradation signals.

This research topic proposes a prognostics model for *multi-sensor* applications where the multi-stream degradation signals are highly incomplete. The model is based on functional LLS regression in which the predictor is a set of multi-stream degradation signals and the response is TTF. LLS regression models have been widely used in reliability engineering and survival analysis. They include a variety of TTF distributions, such as (log)-normal, (log)-logistics, smallest extreme value, and Weibull. The estimation of a functional LLS regression model is usually an intractable problem. As a result, we first use multivariate FPCA to fuse the multistream signals. This enables us to transform the functional regression framework to a classic LLS regression model, in which the predictor is the fused features (known as "FPC-scores") from multivariate FPCA and the response is TTF.

Multivariate FPCA is capable of capturing the joint variation of multi-stream functional data (degradation signals in our case). To estimate the FPC-scores, all the existing estimation methods assume that signals are complete, that is, they are observed continuously and frequently at regular time grids. To be specific, the complete signals from different sensors

are first concatenated to form a signal matrix. Next, Singular Value Decomposition (SVD) is applied to the signal matrix (or equivalently, Eigen Decomposition, ED, is applied to the covariance matrix of the signal matrix) to compute singular (eigen) vectors. Finally, FPC-scores are estimated by projecting signals to the singular vectors. For incomplete signals, however, none of the existing estimation method can work. This is because when the signal matrix is incomplete, neither SVD nor ED can be used to compute the singular vectors. To address this challenge, two algorithms are developed in this topic. The first algorithm, called *subspace detection*, first extracts a basis of the subspace that the degradation signals lie in, by utilizing the incomplete observations. Next, with the help of the basis, a novel feature extraction algorithm is developed to compute the singular vectors of the signal matrix. Finally, FPC-scores are calculated using the singular vectors and the incomplete signals. The second algorithm, known as *signal recovery*, begins with recovering the degradation signals from each sensor via its incomplete observations. Next, the recovered signals from different sensors are concatenated. To address the computational challenge when the concatenated signal matrix is big, we develop an incremental SVD algorithm, which computes the singular vectors of the concatenated signal matrix by adding one of its columns at a time. Finally, FPC-scores are computed using the incomplete signals and the singular vectors.

1.3.5 A supervised dimension reduction-based prognostics model for applications with incomplete signals and censored failure times

In this thesis, Chapter 4 considers applications with incomplete degradation signals from a single sensor and Chapter 5 focuses on modeling incomplete degradation signals from multiple sensors. In addition to incomplete degradation signals, in reality, another challenge stems from the fact that historical failure times are usually *censored*. This is because equipment usually gets replaced or maintained before a failure happens, and thus no failure can be observed. Another possible reason is that there are often constraints on the length

of life tests, and thus data has to be analyzed before all units have failed. In Chapter 6, we develop a prognostics methodology for multi-sensor applications with (highly) incomplete degradation signals and censored historical failure times. This is achieved by proposing *supervised dimension reduction (SDR)*-based prognostic methodology. The model builds an optimization problem that combines a feature extraction term and a regression term. The feature extraction term focuses on extracting low-dimensional features using multi-stream incomplete degradation signals. It works by decomposing each system's degradation signal as a weighted combination of some unknown basis. The weights are known as the features of that system. The second term regresses the fused features against the censored TTFs via LLS regression. The weights and basis in the first term, and the regression parameters in the second term, are estimated simultaneously from the historical dataset by solving the optimization problem mentioned earlier. Since the feature extraction process is supervised by TTFs, it is guaranteed that the extracted features are most informative for TTF prediction. To solve the optimization problem, we develop a Block Prox-Linear Coordinate Descent algorithm, which works by cyclically optimizing a block of variables at each iteration while keeping other blocks fixed. In addition, we theoretically prove the global convergence property of the algorithm.

1.3.6 Residual useful lifetime prediction using a degradation image stream

There is a growing trend in using image sensors, such as infrared and charge coupled device (CCD) cameras, for condition monitoring. This results in a degradation image stream containing rich information about the performance of a system over time. This research topic develops a prognostic model that employs degradation image streams to predict the RUL of systems. Image streams have been extensively used for process monitoring and diagnostics. However, there is little research in the literature focusing on prognostics using image streams. This is mainly due to analytical challenges caused by the complex structure of image streams. One of the key challenges is ultrahigh dimensionality: In addition

to the large number of pixels in each image, the number of images linearly grows over time as new images are recorded. Another challenge is complex spatial-temporal structure. For example, pixels within an image are spatially correlated, and corresponding pixels are temporally correlated across sequential images.

To address the aforementioned challenges, the prognostic methodology in this topic is formulated as a LLS tensor regression model, in which the TTF is treated as the response and degradation image streams as covariates. To model the *spatio-temporal structure* of degradation images, the regression model treats each image stream as a *tensor*, which is defined as a *multidimensional array*. For example, a one-order tensor is a vector, a two-order tensor is a matrix, and objects of order three or higher are called high-order tensors. Degradation image streams constitute a three-order tensor, in which the first two dimensions capture the spatial structure of a single image while the third dimension is used to model the temporal structure of an image stream. One benefit of treating degradation images as a tensor is that it does not break the *spatio-temporal structure* within and between images resulting in more accurate prediction models.

As mentioned earlier, the high dimensionality of degradation image streams poses a significant challenge for parameter estimation as well as computation. As an example if a proper dimensionally method is not applied, to fit a tensor-regression model for a sequence of 20-by-20 images with the length of 50, a three-order tensor coefficient with 20,000 elements should be estimated. To address this challenge, we build scalable estimation methods that reduce the number of parameters by taking the advantage of the fact that although image streams are high-dimensional their essential information can be captured in a low-dimensional space. First, the images streams (tensor) are projected to a low-dimensional tensor subspace that is able to preserve their information. This can be done via applying some dimensionality reduction techniques, such as multilinear principal component analysis [21]. Next, the coefficient tensor corresponding to the projected image tensors are decomposed using two popular tensor decompositions, namely, CP and Tucker.

The CP decomposition decomposes a high-dimensional coefficient tensor as a product of several low-rank basis matrices, and Tucker decomposition expresses it as a product of a low-dimensional core tensor and several factor matrices. Therefore, instead of estimating the coefficient tensor, we only estimate its corresponding core tensors and factor/basis matrices, which helps significantly reduce the computational complexity and the required sample size. Two scalable block relaxation algorithms are developed for model estimation that can achieve a global convergence to a stationary point.

1.4 Dissertation organization

The organization of the dissertation is shown in Figure 1.6. Chapter 1 presents the research background, motivation, data characteristics and challenges, and research topics in this dissertation. Chapter 2 develops a three-step multi-sensor prognostic methodology, which is able to systematically select the informative sensors, fuse the signals from these sensors, and predict residual useful lifetimes of partially degraded systems. Chapter 3 presents a scalable semi-parametric prognostics model specifically designed for large-scale degradation datasets. Two algorithms are developed for signal fusion and one scalable procedure is proposed for model estimation. In Chapter 4, we introduce an adaptive functional regression-based model that uses incomplete degradation signals from a single sensor to predict the remaining useful lifetimes. Chapter 5 introduces a robust prognostic model that is capable of modeling highly incomplete multi-stream degradation signals. Chapter 6 develops a novel supervised dimension reduction-based prognostics model for applications with incomplete multi-stream signals and censored historical failure times. Chapter 7 discusses a new methodology for RUL prediction of a system using a sequence of degradation images. Finally, Chapter 8 concludes this dissertation and discusses future research opportunities.

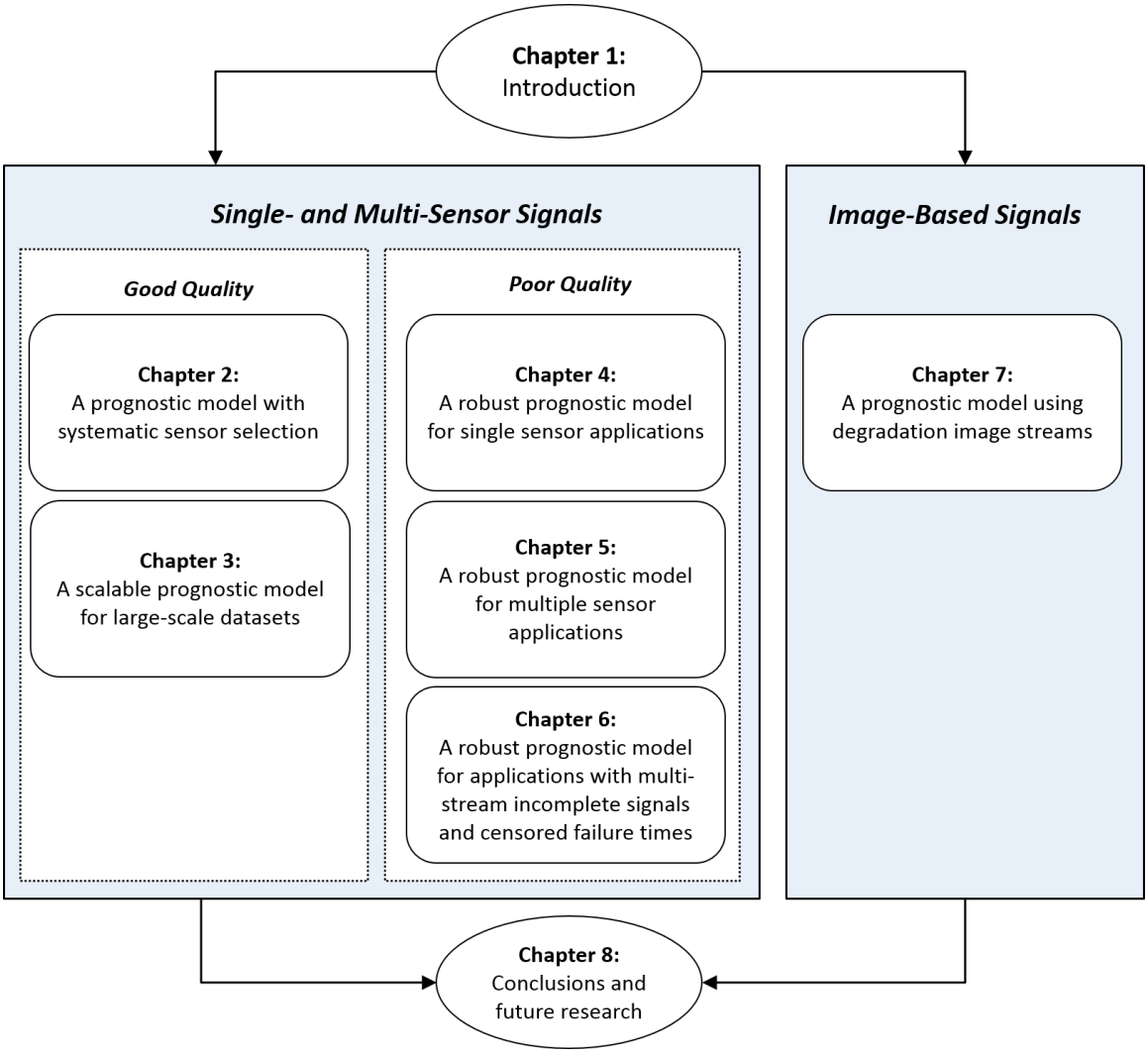


Figure 1.6: Outline of dissertation.

CHAPTER 2
MULTISTREAM SENSOR FUSION-BASED PROGNOSTICS MODEL FOR SYSTEMS WITH SINGLE FAILURE MODES

2.1 Introduction

This chapter focuses on developing a prognostic methodology for engineering systems being monitored by multiple sensors. Multiple sensors are used to capture different aspects of the failure process. Raw signals from these sensors are often transformed into degradation signals that can be used to predict residual useful lifetime. Most of the existing literature focuses either on modeling an individual degradation signal or on combining all the degradation signals together through some sort of fusion mechanism. There are various types of prognostic models that focus on single sensor applications. Examples include using random coefficients models [1, 2], Brownian motion process [3, 4, 5], Gamma process [6, 7, 8], and Markov chains [9, 10]. The second category are the models focusing on multi-sensor settings. They typically rely on combining all the available sensor signals using different types of fusion methods, such as neural networks [22], Hidden Markov models [23, 24], neuro-fuzzy systems [25, 26, 27, 28, 29], multilayer perceptron networks[30], health index models [31, 32, 33, 34] and FPCA [19].

In many multi-sensor applications, there exists some level of redundancy among the sensors. That is, more than one sensor may capture the same physical effects to a similar degree. In some other instances, signals from some sensors may have little or no relation to the underlying physics, thus compromising the accuracy of predicting failures. As a result, selecting the appropriate sensors before the fusion process may possibly lead to better failure predictability.

This chapter builds on the existing body of literature by proposing a multi-sensor prog-

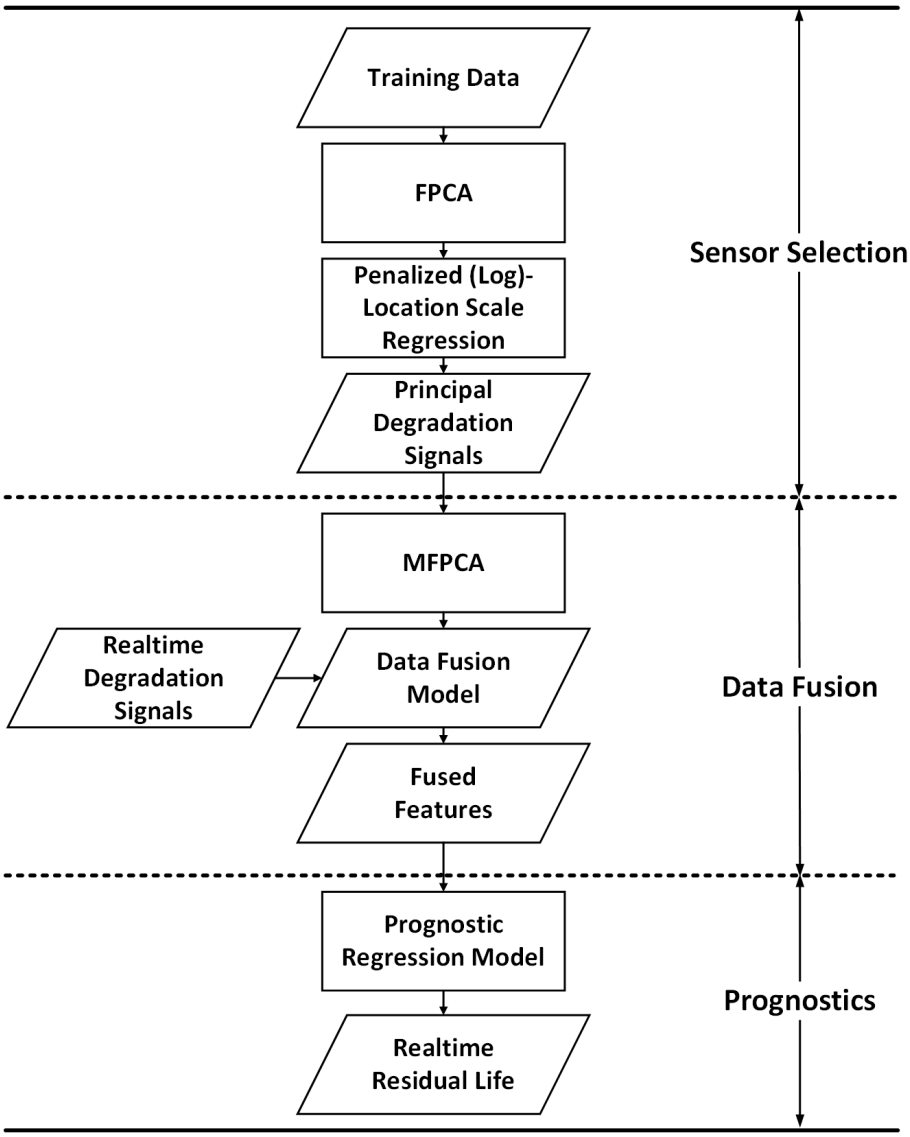


Figure 2.1: Multi-sensor fusion-based prognostics framework.

nostic methodology that incorporates a systematic sensor selection procedure. Very few existing models formally incorporate a sensor selection procedure [31, 32]. Nonetheless, the procedures are based on visual inspection or other subjective procedures, and thus may vary from one user to another. Our methodology consists of three steps shown in Figure 2.1. The first step is a formalized sensor selection algorithm that systematically identifies the most informative sensors that should be used to predict failure and RUL. This step is based upon a penalized variable selection methodology [15, 16]. The goal is to identify highly informative sensors that when combined together provide a relatively comprehen-

sive yet precise characterization of the underlying physical degradation. The penalized variable selection is developed by combining FPCA with a penalized (log)-location-scale functional regression model. FPCA is a popular functional data analysis (FDA) technique that identifies the important sources of patterns and variations among functional data [17]. The degradation signals from each sensor are projected to a low-dimensional feature space spanned by the eigen-functions of the signals' covariance function and provides fused features called FPC-scores. The FPC-scores are then regressed against the TTF to identify the most informative sensors using a penalized (log)-location-scale regression model. (Log)-location-scale regression has been widely used in reliability engineering [35] and can be used with a variety of TTF distributions, such as (log)-normal, (log)-logistics, smallest extreme value, Weibull, etc.

The second step focuses on intelligently combining the degradation signals associated with the informative sensors identified by the sensor selection process. This is achieved by developing a signal fusion algorithm based on Multivariate FPCA (also known as MFPCA) [17]. Multivariate FPCA focuses on capturing the joint variation of multistream functional data. It works by applying ordinary FPCA on the concatenated degradation signals from different sensors. The benefit of using Multivariate FPCA is that it reduces dimensionality of the data and provides fused signal features in the form of MFPC-scores.

Finally, the third step focuses on utilizing the fused signal features for prognostics. This will be accomplished by using an adaptive penalized (log)-location-scale regression model, which estimates RUL and provides the means to continuously update these estimates as real-time signals are observed from fielded systems.

The remainder of this chapter is organized as follows. In Section 2.2 we present the degradation modeling and sensor selection methodology used for identifying the most informative sensors whose signals will be used for prognostics. We then discuss the multistream data fusion algorithm in Section 2.3. Section 2.4 discusses the development of the prognostic model used for estimating and updating RULs of fielded engineering systems.

The accuracy of the prognostic model is evaluated using a simulation study in Section 2.5 and aircraft turbofan engine degradation data from a physics-based simulation model developed by NASA in Section 5.6. Finally, the conclusion and future research directions are presented in Section 7.8.

2.2 Signal model and sensor selection

This framework focuses on systems whose failure is dominated by a single degradation process, which is being monitored by multiple sensors. We assume that raw signals from each individual sensor can be easily synthesized into degradation-based signals. Thus, each sensor has a corresponding degradation signal. Furthermore, we assume that there exists a historical database of degradation signals, a training dataset, that can be leveraged in model estimation. Ideally this database will contain high quality degradation signals from a set of (identical) systems along with their corresponding TTF. The underlying premise of our multi-sensor prognostic methodology is that it is possible to identify a select subset of sensors that provides similar (and sometimes better) characterization of the degradation process, rather than relying on all the sensors used in monitoring. The benefits of doing this include a possible improvement in the accuracy of failure predictability as well as potential reduction in the costs of data acquisition and processing.

A variable selection methodology is utilized to develop our sensor selection procedure. At a high level, this is achieved by combining penalized LLS regression with FPCA to identify the sensors the that are most correlated with the underlying degradation process and ultimately the system's TTF. We consider a training dataset of degradation signals for N systems where each system is monitored by P sensors. Let $s_p(t)$ for $p = 1, 2, ..., P$ denote the observed degradation signal of sensor p, such that $s_p(t)$ are independent noisy realizations of a smooth random function $x_p(\cdot)$ in a bounded time domain [0, T] with unknown mean function $\mathbb{E}[x_p(t)] = \mu_p(t)$ and covariance function $C_p(t, t') = Cov(x_p(t), x_p(t'))$.

We express the degradation signal from sensor p of system i as follows:

$$s_{i,p}(t) = x_{i,p}(t) + \epsilon_{i,p}(t); \quad i = 1, 2, ..., N, \tag{2.1}$$

where $\epsilon_{i,p}(t)$ are assumed to be independent and identically distributed (i.e., i.i.d.) errors with mean zero and variance σ_p^2, and $x_{i,p}(t)$ and $\epsilon_{i,p}(t)$ are independent.

Using Mercer's theorem [36], the covariance matrix $C_p(t, t')$ can be expanded as follows;

$$C_p(t, t') = \sum_{k=1}^{\infty} \lambda_{k,p} \phi_{k,p}(t) \phi_{k,p}(t'), \ t, t' \in [0, T], \tag{2.2}$$

where $\phi_{k,p}(t)$ for $k = 1, 2, \ldots$ are the orthogonal eigen-functions and $\lambda_{1,p} \geq \lambda_{2,p} \geq \ldots$ are the ordered nonnegative eigen-values. Using these eigen-functions, $s_{i,p}(t)$ can be expressed as linear combinations of the orthogonal basis functions as follows:

$$s_{i,p}(t) = \mu_p(t) + \sum_{k=1}^{\infty} \xi_{i,k,p} \phi_{k,p}(t) + \epsilon_{i,p}(t), \tag{2.3}$$

where $\xi_{i,k,p}$ for $k = 1, 2, \ldots,$ are known as FPC-scores, which are uncorrelated random variables with mean zero and variance $\mathbb{E}(\xi_{i,k,p}^2) = \lambda_{k,p}$. Generally, the eigenvalues $\lambda_{k,p}$ for $k = 1, 2, \ldots,$ decrease rapidly and only a small number of eigenvalues would suffice to capture most information of multistream signals and the rest are approximately zero. Therefore, it is often sufficient to use only the eigenfunctions corresponding to significantly nonzero eigenvalues to accurately approximate the signals. Consequently, $s_{i,p}(t)$ can be approximated as shown below;

$$s_{i,p}(t) = \mu_p(t) + \sum_{k=1}^{K_p} \xi_{i,k,p} \phi_{k,p}(t) + \epsilon_{i,p}(t). \tag{2.4}$$

where K_p is the number of significantly nonzero eigenvalues. The value of K_p can usually be chosen by using cross validation (CV) and the Akaike information criterion (AIC) [19]. In practice, the fraction of variance explained (FVE) is another efficient way to determine

K_p (see [37] for more details).

2.2.1 Sensor selection methodology

Based on the premise that the TTF of a system can be predicted by its degradation signals, we establish the following (log)-location-scale regression model for modeling the TTF (i.e., f_i) as a function of the degradation signals (i.e., $\{s_{i,p}(t)\}_{p=1}^{P}$):

$$Pr(y_i \leq y) = \Omega\left(\frac{y - \pi(s_{i,p}(t))}{\sigma}\right), \tag{2.5}$$

where $y_i = f_i$ if f_i follows a location-scale distribution and $y_i = \log(f_i)$ if f_i follows a log-location-scale distribution. Ω is the cumulative distribution function (i.e., cdf) of the location-scale distribution, σ is the scale parameter, and $\pi(s_{i,p}(t))$ is the location parameter, which is a function of some explanatory variables. Here we assume that $\pi(s_{i,p}(t))$ is a function of degradation signals, i.e., $\pi(s_{i,p}(t)) = \alpha_0 + \sum_{p=1}^{P}\int_0^T \alpha_p(t)s_{i,p}(t)dt$, where α_0 is the intercept, $\alpha_p(t)$ is the coefficient function.

The (log)-location-scale regression model in Equation (4.6) can be solved using maximum likelihood estimation (MLE), which yeilds the following optimization criterion:

$$\max_{\alpha_0,\alpha_p(t),\sigma}\left\{-N\log(\sigma) + \sum_{i=1}^{N}\omega\left(\frac{y_i - \alpha_0 - \sum_{p=1}^{P}\int_0^T \alpha_p(t)s_{i,p}(t)dt}{\sigma}\right)\right\}, \tag{2.6}$$

where $\omega(\cdot) = \log w(\cdot)$ and $w(\cdot)$ is the probability density function (i.e., pdf) of the location-scale distribution. For example, $\omega(x) = 1/\sqrt{2\pi}\exp(-x^2/2)$ for normal distribution, $\omega(x) = \exp(x - \exp(x))$ for smallest extreme value distribution and $\omega(x) = \exp(x)/(1 + \exp(x))^2$ for logistic distribution. Criterion (3.13) is nonconvex, which causes computational problems. To address this challenge, we apply the following re-parameterization: $\tilde{\sigma} = 1/\sigma, \tilde{y}_i = y_i/\sigma, \tilde{\alpha}_0 = \alpha_0/\sigma$ and $\tilde{\alpha}_p(t) = \alpha_p(t)/\sigma$. As a result, criterion (3.13) turns

into the following criterion:

$$\max_{\tilde{\alpha}_0,\tilde{\alpha}_p(t),\tilde{\sigma}} \left\{ N\log(\tilde{\sigma}) + \sum_{i=1}^{N} \omega \left(\tilde{y}_i - \tilde{\alpha}_0 - \sum_{p=1}^{P} \int_0^T \tilde{\alpha}_p(t) s_{i,p}(t) dt \right) \right\}, \tag{2.7}$$

Sensor selection is performed by incorporating a variable selection procedure into the MLE criterion expressed in (2.7). Classical variable selection techniques like stepwise regression are often computationally intensive and sometimes unstable. On the other hand, regularization-based techniques like non-negative garrote (NNG) [38], LASSO [39], SCAD [40], elastic net [41], and adaptive LASSO [42] exhibit some advantages with respect to selection stability and prediction accuracy. In this work, we use group non-negative garrote (GNNG) penalty [43] to penalize the group of coefficients corresponding to each degradation signal, i.e., $\{\alpha_p(t)\}$. Consider the following parameterization, $\tilde{\alpha}_p(t) = \hat{\alpha}_p(t) d_p$, where $\hat{\alpha}_p(t)$ is a known weight function and $d_p \geq 0$ is the shrinking factor for signal p, then based on criterion (2.7), the penalized maximum log-likelihood function using the GNNG can be expressed as follows;

$$\max_{\tilde{\alpha}_0,d_p,\tilde{\sigma}} \left\{ N\log(\tilde{\sigma}) + \sum_{i=1}^{N} \omega \left(\tilde{y}_i - \tilde{\alpha}_0 - \sum_{p=1}^{P} \int_0^T \hat{\alpha}_p(t) d_p s_{i,p}(t) dt \right) - \lambda \sum_{p=1}^{P} d_p \right\}, \text{ s.t. } d_p \geq 0, \tag{2.8}$$

where λ is the tuning parameter and d_p represents the importance of the group (or sensor) p.

Selecting the most informative sensors is accomplished by optimizing criterion (2.8). Hereafter, the degradation signals associated with the selected sensors will be referred to as the principle degradation signals (PD-signals). Any sensor whose corresponding d_p is non-zero is considered informative (important), and its corresponding degradation signal is designated as one of the PD-signals that will be used for prognostics, estimating the RUL of the system. Recall that the eigen-functions of the covariance functions of $\{x_{i,p}(t)\}_{i=1}^{N}$ form a complete orthogonal basis, which means that the functions $\hat{\alpha}_p(t)$ can be expanded

as $\hat{\alpha}_p(t) = \sum_{k=1}^{\infty} \beta_{k,p}\phi_{k,p}(t)$, where $\beta_{k,p}$ is the coefficient. Similarly, since the eigen-values $\lambda_{k,p}$ for $k = 1, 2, \ldots$ decrease rapidly (as mentioned earlier), the expansion can be truncated by using the first K_p terms. Thus, we get $\hat{\alpha}_p(t) = \sum_{k=1}^{K_p} \beta_{k,p}\phi_{k,p}(t)$. Substituting $s_{i,p}(t)$ and $\hat{\alpha}_p(t)$ into the location parameter in criterion (2.8) yields the following expression (the details of the derivation are given in Appendix A):

$$\pi(s_{i,p}(t)) = \tilde{\alpha}_0 + \sum_{p=1}^{P} \int_0^T \hat{\alpha}_p(t) d_p s_{i,p}(t) dt = \beta_0 + \sum_{p=1}^{P} d_p \sum_{k=1}^{K_p} \beta_{k,p}\xi_{i,k,p}, \tag{2.9}$$

where $\xi_{i,k,p}$ is the FPC-score defined earlier in Equation (2.4). Consequently, The penalized maximum likelihood function criterion expressed in (2.8) can be rewritten in the following form;

$$\max_{\beta_0, d_p, \tilde{\sigma}} \left\{ N \log(\tilde{\sigma}) + \sum_{i=1}^{N} \omega \left(\tilde{y}_i - \beta_0 - \sum_{p=1}^{P} d_p \sum_{k=1}^{K_p} \beta_{k,p}\xi_{i,k,p} \right) - \lambda \sum_{p=1}^{P} K_p d_p \right\}, \text{ s.t.} d_p \geq 0, \tag{2.10}$$

where $\beta_{k,p}$ is the vector of the weights associated with sensor p. Generally, we can use maximum likelihood, lasso, or ridge estimates of $\beta_{k,p}$ as the weights [15]. As mentioned earlier, the sensors with $\hat{d_p} > 0$ are selected, and their corresponding signals are designated as the PD-Signals.

2.2.2 Model estimation

The optimization model expressed in Equation (2.10) can be solved using the coordinate descent algorithm used in [44]. However, this requires first estimating the FPC-scores, $\xi_{i,k,p}$, which can be achieved using the training dataset. To do this, we first estimate the mean $\mu_p(t)$ and covariance functions $C_p(t, t')$ for $p = 1, 2, \ldots, P$. To simplify notation, and without loss of generality, we assume that the sampling frequency for all sensors is the same.

Local linear regression is used to estimate $\mu_p(t)$ [45, 46]. The estimated mean function

for signal p, which we denote as $\hat{\mu}_p(t)$, can be obtained by minimizing the following loss function,

$$\min_{a_p,b_p} \sum_{i=1}^{N} \sum_{j=1}^{J_i} W(\frac{t_j - t}{w_p})\{s_{i,p}(t_j) - a_p - b_p(t - t_j)\}^2, \tag{2.11}$$

where $\{t_j\} \in [0, T]$ for $j = 1, \ldots, J_i$, denotes discrete observation time points, and J_i represents the length of degradation signals in system i. w_p is the smoothing bandwidth, which is selected by using the one-curve-leave-out cross-validation method presented in [47], and $W(\cdot)$ is a Gaussian kernel function. The estimated mean function is $\hat{\mu}_p(t) = \hat{a}_p$, where $\hat{a}_p$ is the solution to Equation (C.2).

The covariance function, $C_p(t, t')$, is also estimated using local linear regression[45, 46]. From Equation (2.1), we can see that $Cov(s_p(t_j), s_p(t_k)) = Cov(x_p(t_j), x_p(t_k)) + \sigma^2\delta_{j,k}$, where $\delta_{j,k} = 1$ if $t_j = t_k$ and 0 otherwise. Let $G_{i,p}(t_j, t_k) = (s_{i,p}(t_j) - \hat{\mu}_p(t_j))(s_{i,p}(t_k) - \hat{\mu}_p(t_k))$ be the "raw" covariances of degradation signal p of system i, we have $E[G_{i,p}(t_j, t_k)] = Cov(x_p(t_j), x_p(t_k)) + \sigma^2\delta_{j,k}$. In other words, the noise term only lies on the diagonal elements of the raw covariances. Therefore, the diagonal elements of $G_{i,p}(t_j, t_k)$ are removed and only off-diagonal elements of the raw covariances are considered for estimation [48]. The covariance function is estimated by minimizing the following loss function;

$$\min_{a_p',b_p',c_p'} \sum_{i=1}^{N} \sum_{1 \le j \ne k \le J_i} W(\frac{t_j - t}{w_p'}, \frac{t_k - t'}{w_p'})\{G_{i,p}(t_j, t_k) - a_p' - b_p'(t - t_j) - c_p'(t' - t_k)\}^2, \tag{2.12}$$

where w_p' is the smoothing bandwidth, and $W(\cdot)$ is a bivariate Gaussian kernel function. The estimated covariance function is obtained as $\hat{C}_p(t, t') = \hat{a_p'}$, for $t, t' \in [0, T]$ where $\hat{a'}_p$ is the solution to Equation (C.3).

Once the covariance function has been estimated, our next step is to estimate its eigen-functions $\phi_{k,p}(t)$ and eigen-values $\lambda_{k,p}$. This is achieved by solving the following eigen-equations:

$$\int_0^T \hat{C}_p(t, t')\hat{\phi}_{k,p}(t)dt = \hat{\lambda}_{k,p}\hat{\phi}_{k,p}(t'), \tag{2.13}$$

where $\int_0^T \hat{\phi}_{k,p}(t)\hat{\phi}_{m,p}(t)dt = 1$ if $k = m$ and 0 otherwise.

Equation (2.13) can be solved by discretizing the estimated covariance surface $\hat{C}_p(t, t')$ [47]. The FPC-scores are estimated using $\hat{\xi}_{i,k,p} = \int_0^T (s_{i,p}(t) - \hat{\mu}_p(t))\hat{\phi}_{k,p}(t)dt$, which is approximated numerically by $\hat{\xi}_{i,k,p} = \sum_{j=0}^{J_i}((s_{i,p}(t_j) - \hat{\mu}_p(t_j))\hat{\phi}_{k,p}(t_j)(t_j - t_{j-1}))$; $t_0 = 0$. Finally, the estimated $\hat{\xi}_{i,k,p}$ can then be substituted in the penalized maximum likelihood criterion defined in Equation (2.10) to estimate the importance of each sensor, i.e., d_p. This allows us to identify the most informative sensors. Recall that the degradation signals associated with these select sensors are referred to as the PD-signals. These form the basis for predicting the RULs of partially degraded systems.

2.3 Multistream signal fusion model

In this section, we focus on how to efficiently combine the PD-signals such that their cross-correlations can be leveraged to provide accurate predictions of RUL. To do this, we develop a multistream signal fusion methodology based on Multivariate FPCA. Multivariate FPCA is an extension of the FPCA framework. Whereas FPCA focuses on identifying the important sources of variation among a single type of functional data, Multivariate FPCA focuses on capturing the joint variation of multistream functional data. One of the key benefits of using Multivariate FPCA is that it provides a way to characterize the cross-correlation between the PD-signals measured by different sensors. By modeling the relationship between different informative sensors, we can better characterize the underlying degradation process, and therefore improve prediction accuracy. Another advantage of using Multivariate FPCA is that it reduces the dimensionality of the PD-signals, and provides fused features (in the form of MFPC-scores) that become the predictors in our subsequent prognostic degradation modeling framework.

2.3.1 Signal fusion using Multivariate FPCA

Based on [17], Multivariate FPCA works by first concatenating different sources of functional data into a single vector. Next, FPCA is applied to the concatenated vector in a

conventional manner. Note that signal fusion is only applied to the PD-signals associated with the sensors identified by the sensor selection methodology. We begin by assuming that the PD-signals of unit i are independent noisy realizations of an M-dimensional stochastic process $\boldsymbol{x}(\cdot)$ in a bounded time domain $[0, T]$ with unknown mean function $\boldsymbol{\mu}(t)$ and covariance function $\boldsymbol{C}(t, t')$, where M is the number of selected sensors. Here, $\boldsymbol{\mu}(t)$ represents the combined underlying deterministic trend of the degradation process, and $\boldsymbol{C}(t, t')$ represents the deviation from the underlying degradation trend due to system-to-system degradation variability. Therefore, the PD-signals for system i can be expressed as follows:

$$\boldsymbol{s}_i(t) = \boldsymbol{x}_i(t) + \boldsymbol{\epsilon}_i(t), \tag{2.14}$$

for $i = 1, \ldots, N$, where N is the system number; $\boldsymbol{s}_i(t) = (s_{i,1}(t), \ldots, s_{i,M}(t))^\top$, $\boldsymbol{x}_i(t) = (x_{i,1}(t), \ldots, x_{i,M}(t))^\top$ and $\boldsymbol{\epsilon}_i(t) = (\epsilon_{i,1}(t), \ldots, \epsilon_{i,M}(t))^\top$; $\boldsymbol{x}_i(t)$ and $\boldsymbol{\epsilon}_i(t)$ are assumed to be independent. Note that the covariance function $\boldsymbol{C}(t, t') = \mathbb{E}[(\boldsymbol{x}(t) - \boldsymbol{\mu}(t))(\boldsymbol{x}(t') - \boldsymbol{\mu}(t'))^\top]$, $t, t' \in [0, T]$ is an $M \times M$ block matrix with the following form;

$$\boldsymbol{C}(t, t') = \begin{pmatrix} C_{1,1}(t, t') & \ldots & C_{1,M}(t, t') \\ \vdots & \ddots & \vdots \\ C_{M,1}(t, t') & \ldots & C_{M,M}(t, t') \end{pmatrix}, \tag{2.15}$$

with $C_{g,h}(t, t') = Cov(x_g(t), x_h(t'))$, for $g = 1, \ldots, M$ and $h = 1, \ldots, M$.

Mercer's theorem [36] is also used to decompose the covariance function $\boldsymbol{C}(t, t')$ as follows;

$$\boldsymbol{C}(t, t') = \sum_{k=1}^{\infty} \eta_k \boldsymbol{\psi}_k(t) \boldsymbol{\psi}_k(t')^\top, \tag{2.16}$$

where $\boldsymbol{\psi}_k(t) = (\psi_{k,1}(t), \ldots, \psi_{k,M}(t))^\top$ for $k = 1, 2, \ldots,$ are the eigenfunctions and $\eta_1 \geq$

$\eta_2 \geq \ldots$, are the ordered nonnegative eigenvalues. Hence, $\boldsymbol{x}_i(t)$, can be rewritten as,

$$\boldsymbol{x}_i(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{\infty} \zeta_{i,k} \boldsymbol{\psi}_k(t) \tag{2.17}$$

where $\zeta_{i,k} = \int_0^T (\boldsymbol{x}_i(t) - \boldsymbol{\mu}(t))^\top \boldsymbol{\psi}_k(t) dt = \sum_{m=1}^M \int_0^T (x_{i,m}(t) - \mu_m(t)) \psi_{k,m}(t) dt$ for $k = 1, 2, \ldots,$ are the MFPC-scores. These scores are independent random variables with mean $\mathbb{E}[\zeta_{i,k}] = 0$ and variance $\mathbb{E}[\zeta_{i,k}^2] = \eta_k$. The signal model expressed in (2.14) can now be expressed as,

$$\boldsymbol{s}_i(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{K} \zeta_{i,k} \boldsymbol{\psi}_k(t) + \boldsymbol{\epsilon}_i(t) \tag{2.18}$$

where similar to the decomposition in Section 2.2, the model is truncated by choosing the first K eigenvalues and K is also chosen using CV, AIC or FVE criterion (see Equation (2.4)).

2.3.2 Estimating the fused signal features

Given a historical training dataset of signals, we can estimate the signal model expressed in Equation (2.18). Let $\{t_j\}$ for $j = 1, \ldots, J$, where $t_j \in [0, T]$ denote discrete observation time points, where J is the total number of observations for each system. We first estimate the mean function $\boldsymbol{\mu}(t)$, where $\boldsymbol{\mu}(t) = (\mu_1(t), \ldots, \mu_M(t))^\top$. The mth element of the mean function $\boldsymbol{\mu}(t)$ is estimated using $\hat{\mu}_m(t_j) = \frac{1}{N} \sum_{i=1}^N s_{i,m}(t_j)$ for $m = 1, \ldots, M$, where N is the system number.

To estimate the covariance matrix, we note that $\boldsymbol{C}(t, t')$ is a $M \times M$ block matrix. Each block $C_{g,h}(t, t')$ is estimated individually using $\hat{C}_{g,h}(t_j, t_k) = \frac{1}{N-1} \sum_{i=1}^N (s_{i,g}(t_j) - \hat{\mu}_g(t_j))(s_{i,h}(t_k) - \hat{\mu}_h(t_k))$, for $g = 1, \ldots, M$, $h = 1, \ldots, M$, $j = 1, \ldots, J$ and $k = 1, \ldots, J$. Since $\boldsymbol{C}(t, t')$ is a symmetric matrix, we only need to estimate $C_{g,h}(t, t')$ with $g \geq h$.

Next, we estimate the MFPC-scores ζ_{ik}. Recall that $\boldsymbol{C}(t, t') = \sum_{k=1}^{\infty} \eta_k \boldsymbol{\psi}_k(t) \boldsymbol{\psi}_k(t')^\top$, where the eigenfunctions $\boldsymbol{\psi}_k(t)$ and the eigenvalues η_k can now be estimated by solving

the following eigen equations:

$$\int_0^T \hat{\boldsymbol{C}}(t,t')\hat{\boldsymbol{\psi}}_k(t)dt = \hat{\eta}_k\hat{\boldsymbol{\psi}}_k(t'), \tag{2.19}$$

where $\int_0^T \hat{\boldsymbol{\psi}}_k(t)\hat{\boldsymbol{\psi}}_l(t)dt = 1$ if $l = k$ and 0 otherwise. Equation (C.4) is solved by discretizing the estimated covariance surface $\hat{\boldsymbol{C}}(t,t')$ (details can be found in [47]). In other words, the MFPC-scores can be estimated using numerical integration where $\hat{\zeta}_{i,k} = \sum_{m=1}^{M}\sum_{j=1}^{J_i}(s_{i,m}(t_{j,m}) - \hat{\mu}_m(t_{j,m}))\hat{\phi}_{k,m}(t_{j,m})(t_{j,m} - t_{j-1,m}))$, and where $t_0 = 0$. Note that the MFPC-scores represent the collective (reduced) features of the PD-signals, which will be used to estimate the RUL of the system.

2.4 Residual useful lifetime prediction and real-time updating

Predicting the RUL is accomplished by developing a (log)-location-scale regression model between the system TTFs and the PD-signals estimated earlier. To do this, we let $\boldsymbol{s}_i(t)$ denote a vector of PD-signals and f_i the corresponding TTF of system *i*. The (log)-location-scale regression model relating these two factors is expressed below;

$$Pr(y_i \leq y) = \Omega\left(\frac{y - \rho_0 - \int_0^T \boldsymbol{\rho}(t)^\top \boldsymbol{s}_i(t)dt}{\sigma}\right), \tag{2.20}$$

where $y_i = f_i$ if f_i follows a location-scale distribution and $y_i = \log(f_i)$ if f_i follows a log-location-scale distribution; ρ_o is the intercept; and $\boldsymbol{\rho}(t) = (\rho_1(t), \ldots, \rho_M(t))^\top$ is the regression coefficient function. The coefficient funciton can be expanded using the eigen-functions of $\{\boldsymbol{x}_i(t)\}_{i=1}^N$ as $\boldsymbol{\rho}(t) = \sum_{k=1}^{\infty}\theta_k\boldsymbol{\psi}_k(t)$. Similar to the derivation in Appendix A, the location parameter in Equation (2.20) can be expressed as $\boldsymbol{\pi}(\boldsymbol{s}_i(t)) = \rho_0 + \int_0^T \boldsymbol{\rho}(t)^\top \boldsymbol{s}_i(t)dt = \theta_0 + \sum_{k=1}^{K}\theta_k\zeta_{i,k}$, where $\zeta_{i,k}$ are the MFPC-scores defined in Equation (2.17), and θ_0 and θ_k are the coefficients. As a result, Equation (2.20) can be reexpressed

as

$$Pr(y_i \leq y) = \Omega\left(\frac{y - \theta_0 - \sum_{k=1}^{K} \theta_k \zeta_{i,k}}{\sigma}\right), \tag{2.21}$$

The MLE is used to solve the regression model in Equation (2.21), which yields the following optimization criterion:

$$\max_{\theta_0,\theta_k,\sigma}\left\{-N\log(\sigma) + \sum_{i=1}^{N}\omega\left(\frac{y_i - \theta_0 - \sum_{k=1}^{K}\theta_k\zeta_{i,k}}{\sigma}\right)\right\}, \tag{2.22}$$

In order to transfer (2.22) to a convex optimization criterion, we apply the following re-parameterization: $\tilde{\sigma} = 1/\sigma, \tilde{y}_i = y_i/\sigma, \tilde{\theta}_0 = \theta_0/\sigma, \tilde{\theta}_k = \theta_k/\sigma$ (see [37] for more details). In addtion, to improve the prediction accuracy, we use non-negative garrote [38] for penalization, which yields the following penalized maximum likelihood function criterion;

$$\max_{\tilde{\theta}_0,d_k,\tilde{\sigma}}\left\{N\log(\tilde{\sigma}) + \sum_{i=1}^{N}\omega\left(\tilde{y}_i - \tilde{\theta}_0 - \sum_{k=1}^{K} d_k\hat{\theta}_k\zeta_{i,k}\right) - \lambda\sum_{p=1}^{K} d_k\right\}, \text{ s.t. } d_k \geq 0, \tag{2.23}$$

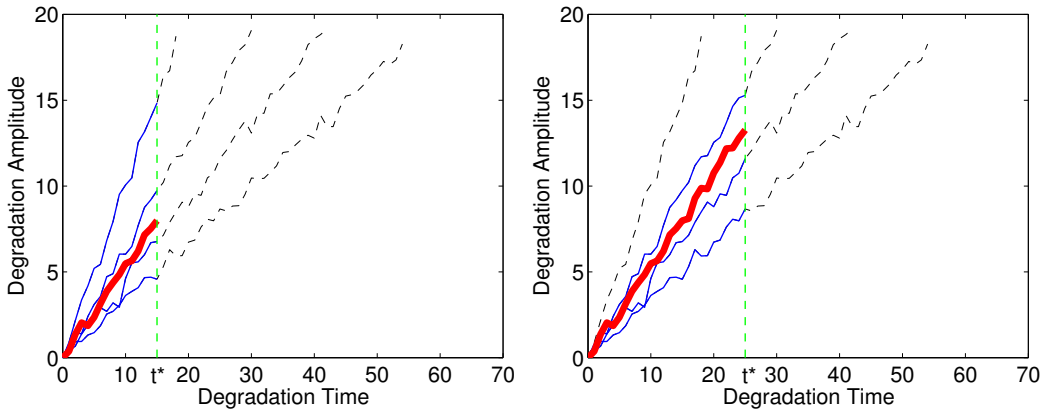
where $\hat{\theta}_k$ is the initial regression coefficients estimated using MLE without penalization, d_k is the shrinkage factor, and λ is the tuning parameter.

Our goal is to predict and update, in near real-time, the RUL of partially degraded systems that are still operating in the field. To do this, PD-signals observed from fielded systems (hereafter referred to as *observed* PD-Signals) are used to update the fused signals features, which are then used to update the predicted RUL. This is accomplished by combining the *observed* PD-signals from a specific (fielded) system with PD-signals of the training dataset, and using our signal fusion algorithm to compute updated MFPC-scores. The updated MFPC-scores are then used revise the predicted lifetime using the (log)-location-scale regression model estimated earlier in Equation (2.23). The RUL is obtained by subtracting the current observation time.

To calculate the scores, Multivariate FPCA requires that the PD-signals from each type of sensor share the same time domain. That is, when combining the PD-signals observed from the field with the PD-signals from the training dataset, signals for each sensor must share the same time domain. We note that the systems of the training sets have different TTFs. Thus, there are no guarantees that for a given sensor, the PD-signals from various systems will share the same time domain. To address this limitation, we leverage recent developments in functional regression. Specifically, we utilize the adaptive functional regression approach proposed by [49].

The basis of adaptive functional regression is that training systems whose lifetime is smaller than the current observation time of the PD-signal are removed from the training dataset. In other words, they are not used to recalculate the MFPC-scores. Only the PD-signals of the training systems whose lifetime is bigger than the current observation time are combined with the PD-signals observed from the fielded system to update the MFPC-scores. The chosen training PD-signals are then truncated at the current observation time. This way all the signals involved share the same time domain as the observed PD-signal. Figure 2.2 illustrates how the adaptive updating approach would apply to a single sensor scenario. In each graph, the dotted lines represent the entire set of training signals. The continuous part marks the part of the data that is used to estimate the MFPC-scores. The signal marked with the thick continuous line represents the portion of the degradation signal of the fielded system that has been observed up to time t^*.

We summarize our prognostic model as follows. Each time new PD-signals are observed from a fielded system, we select the training PD-signals that satisfy the criteria mentioned earlier and truncate them at the current observation time. Next, we apply Multivariate FPCA to the selected PD-signals (the ones selected from the training dataset along with the observed signal), and estimate revised MFPC-scores. A penalized regression model is then defined by setting the revised MFPC-scores as the dependent variables and setting the TTF corresponding to the selected training PD-signals as the dependent variables. Finally,



(a) Updating at time $t^* = 15$ (b) Updating at time $t^* = 25$

Figure 2.2: Method of updating the model as time advances.

the lifetime of the fielded system can be estimated.

The most time-consuming computation in our prognostic model is Multivariate FPCA, which consists of two key operations: covariance matrix estimation and eigen-decomposition. Suppose the system number is N, the selected sensor number is M and the observation number for each sensor is J, the computational complexity of covariance matrix estimation and eigen-decomposition are $\mathcal{O}(NM^2J^2)$ and $\mathcal{O}(M^3J^3)$, respectively. As a result, the complexity of Multivariate FPCA is $\mathcal{O}(NM^2J^2+M^3J^3)$, which is high when M and J are large. This in turn shows the importance of our sensor selection procedure, which often can tremendously reduce the number of sensors (reduce M) involved in the prognostics model. For the case where the number of selected sensor is large, say thousands, a computationally efficient solution can be found in [37].

2.5 Simulation study

In this section, we conduct a simulation study to validate the proposed prognostic methodology. We consider two (log)-location-scale regression distributions commonly used in reliability, the normal and lognormal distributions. For each distribution, we evaluate the performance of our approach in terms of the effectiveness of the sensor selection model

and the accuracy of predicting the RUL. We compare the performance of our methodology, designated "selection," with the case that all sensors are used for predicting RUL, designated "no selection." In both approaches, the number of FPC-scores are chosen so that more than 95% of signal variations are explained by the chosen scores. The tuning parameters of Equations (2.8) and (2.23) are determined using the leave-one-out cross-validation method. Several values for the tuning parameters within an applicable range are used to train the model. Then, the validation error is calculated for different parameter values. Tuning parameter leading to the least mean square error is chosen to be the optimal tuning parameter.

2.5.1 Simulation model

We consider a system monitored by 40 sensors. We assume that only 4 sensors are informative. In other words, the system TTF can be accurately predicted using the degradation signals from these 4 sensors. We also assume that the set of non-informative sensors are divided into two groups. The first group consists of 16 sensors whose degradation signals have a relatively low correlation with the degradation process and therefore the TTF. The second group consists of 20 sensors whose signals are pure noise and exhibit no trends.

Two hundred instances of this system are generated. A randomly subset of 160 systems are chosen for training while the remaining 40 instances are used for testing. For each instance i; $i = 1, \ldots, 200$, we begin by simulating the underlying degradation path of the system using the following functional form; $s_i(t) = -\theta_i / \ln(t)$, where $\theta_i \sim N(1, 0.25^2)$ and $0 \leqslant t < 1$. The TTF is computed as the first time point that the underlying degradation trajectory, $s_i(t)$, crosses the threshold D, where $D = 2$. Next, degradation signals from the 40 sensors are simulated as follows:

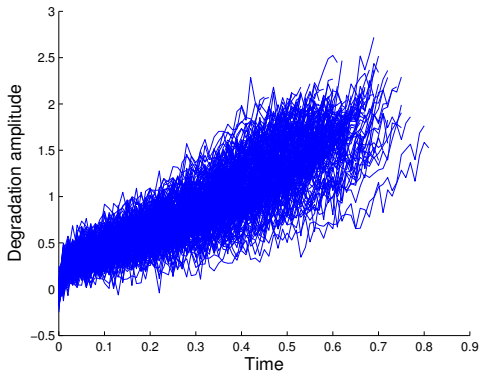
(a) Degradation signals from the informative sensors are generated using the following model $s_{i,p}(t) = -\theta_{i,p} / \ln(t) + \epsilon_{i,p}(t)$, where $p = 1, \ldots, 4$ and $\epsilon_{i,p}(t) \sim N(0, 0.1^2)$. Since informative sensors are highly correlated with the underlying degradation process, we generate $\theta_{i,p}\,; p = 1, ..., 4$ from the following conditional distribution $\theta_{i,p}|\theta_i \sim N(1, 0.25^2)$

such that the correlation between $\theta_{i,p}$ and θ_i is a uniform random number chosen from the interval $[0.8, 0.99]$.

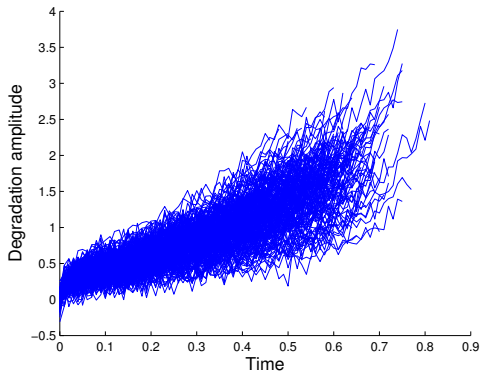
(b) Degradation signals of the non-informative sensors in group 1 (i.e., $s_{i,p}(t);\ p = 5, \ldots, 20$) are also simulated similarly. However, pairwise correlation is randomly chosen from the interval $[0.1, 0.6]$.

(c) Degradation signals of the non-informative sensors in group 2 are simulated from $s_{i,p}(t) = \epsilon_{i,p}(t);\ p = 21, \ldots, 40$ where $\epsilon_{i,p}(t) \sim N(0, 1)$.

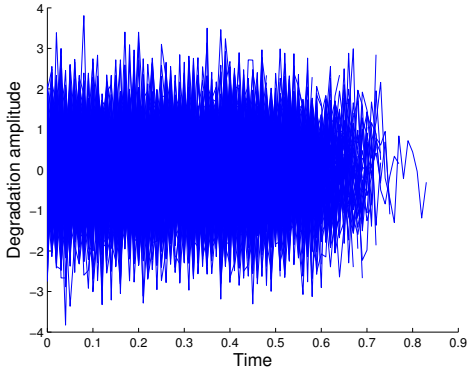
Samples of 200 simulated signals for each sensor type are shown in figure 2.3. The whole simulation procedure is replicated 100 times.



(a) Informative sensors



(b) non-informative sensors (poor correlation)



(c) non-informative sensors (pure noise)

Figure 2.3: Example signals of different type of sensors

2.5.2 Results and analysis

To test the effectiveness of the sensors selection methodology, we begin by applying the sensor selection model discussed in Section 3 to the simulated dataset.

To account for the variability in the length of the signals (discussed in Section 2.4), multiple training subsamples are generated based on the length of signals (or equivalent TTFs). We first, sort TTFs such that $TTF_1 \leq TTF_2 \leq \cdots \leq TTF_S$, where $S \leq 160$. Next, we define subsample j as the systems whose TTFs are greater than or equal to TTF_j, for $j = 1, \ldots, S$. As a case in point, subsample 1 includes all 160 training systems, subsample 2 also includes all training systems excluding the system with the smallest TTFs, and so forth.

The proposed sensor selection method is applied to all of these S subsmaples and the weighted selection rate is computed by using the following equation;

$$r_p = \frac{\sum_{j=1}^{S} I_p N_j}{\sum_{j=1}^{S} N_j} \times 100\%, \tag{2.24}$$

where r_p is the selection rate for sensor p, $I_p = 1$ if sensor p is selected by subsample j and 0 otherwise, and N_s is the system size of subsample j. The selection results for normal and lognormal distributions are reported in Table 2.1 and Table 2.2, respectively.

Table 2.1: Sensor selection results for normal regression model.

	Selected	Dropped
Informative sensors	94.6%	5.4%
Noninformative sensors	2.8%	97.2%

Table 2.2: Sensor selection results for lognormal regression model.

	Selected	Dropped
Informative sensors	96.5%	3.5%
Noninformative sensors	2.3%	97.7%

As shown in the tables above, the lognormal and normal models selected 94.6% and

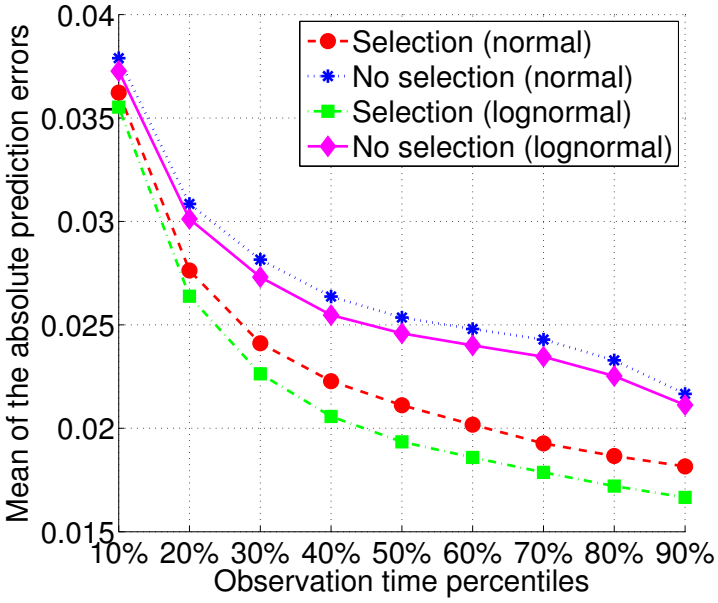
96.5% of informative sensors, respectively. Both lognormal and normal models perform similarly with regards to removing non-informative sensors.

Next, we evaluate the accuracy of predicting the RUL of the simulated test systems using our proposed methodology for the normal and lognormal cases–designated "selection (normal)" and "selection (lognormal)"–and compare it with the benchmark model designated "no selection (normal)" and "no selection (lognormal)". For our methodology, the PD-signals identified by the sensor selection procedure are first combined using the signal fusion algorithms presented in Section 2.3. The fused features (MFPC-scores) are then used for prognostics. Signals from each test system are combined with the training dataset to predict and update its RUL using the time-varying functional regression framework discussed earlier in Section 2.4. Residual lifetimes are evaluated at the following life percentiles: $10\%, 20\%, \ldots, 90\%$, where for example the 10^{th} life percentile implies that 10% of the system's lifetime was attained at the time the prediction was evaluated. The prediction errors are calculated using the following expression:

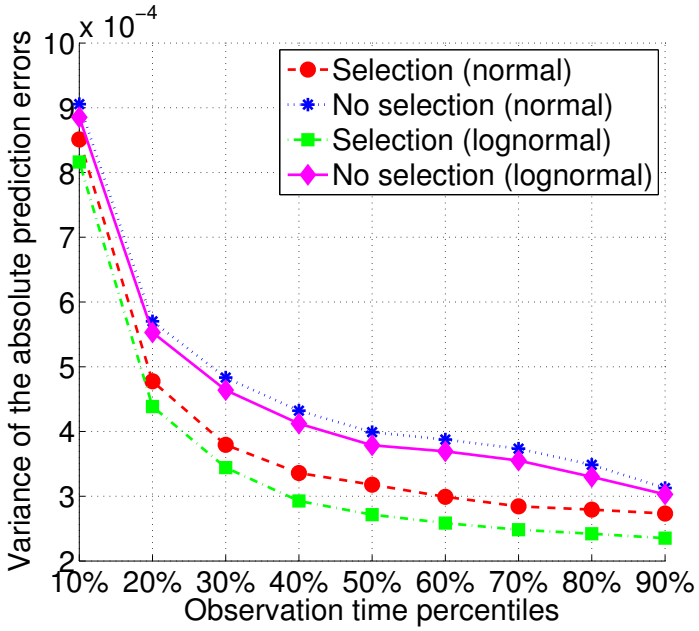
$$\text{Absolute Prediction Error} = \frac{|\text{Estimated Life - Actual Life}|}{\text{Actual Life}}. \tag{2.25}$$

Plot (a) of Figure 2.4 shows the mean of absolute prediction errors and Plot (b) shows the variance of the absolute prediction errors. The two plots confirm that our proposed sensor selection approach has significantly better prediction accuracy (mean) and precision (variance) for both the normal and lognormal scnearios. The improvement is more pronounced at higher life percentiles. For example, at the 70^{th} life percentile, the mean prediction errors for "selection (lognormal)" and "no selection (lognormal)" are 0.018 and 0.023 respectively, which indicates 20% improvement in the prediction accuracy achieved by the sensor selection methodology.

Figure 2.4 also indicated that the lognormal model outperforms the normal model. A possible explanation is that the degradation signals used in this study are generated from



(a) Mean



(b) Variance

Figure 2.4: Mean and variance of the absolute prediction errors.

the following model $s_{i,p}(t) = -\theta_{i,p}/\ln(t) + \epsilon_{i,p}(t)$, in which the logarithmic TTF (i.e., $\log(f_i)$) can be shown to be a linear function of the MFPC-scores of degradation signals (i.e., $\zeta_{i,k}$). Therefore, the lognormal model is supposed to be more suitable than normal model to capture the regression relationship between TTF and MFPC-scores. In reality, we

may use model selection criteria (e.g., AIC, Bayesian Information Criterion) to determine the proper choice of the (log)-location-scale distribution.

2.6 Case study: Aircraft turbofan engine application

In this case study, multi-sensor degradation data from an aircraft turbofan engine is simulated using a physics-based simulator. The dataset, available from [11] is comprised of the following; (1) degradation signals from 100 *training* engines that were run to failure, (2) degradation signals from an additional 100 *test* engines whose operation was prematurely terminated at random time points prior to their failure time, and (3) the real TTFs of the 100 *test* engines. Each engine was monitored using 21 sensors, which are listed in Table 2.3. Since the degradation signals for some of the sensors (1, 5, 6, 10, 16, 18, 19) show no trend, independent and identically distributed white noise are added to them. Figure 2.6 shows the degradation signals of the 21 sensors for the 100 training engines.

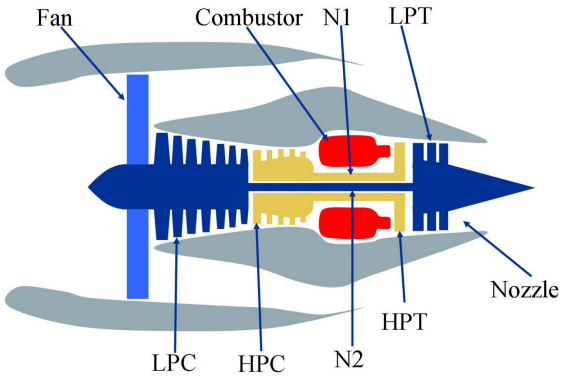


Figure 2.5: Simplified diagram of engine simulated in C-MAPSS [50].

The turbofan engine degradation dataset was simulated using commercial modular aero-propulsion system simulation (C-MAPSS), a simulation module developed in Matlab and Simulink environment. The C-MAPSS can be used for simulating a realistic commercial turbofan engine of $90,000$ lb thrust class [50]. By setting the value of several input parameters and specifying the operation conditions, a user can simulate the effects of faults and deterioration in one or more of the rotating components (Fan, LPC, HPC, HPT and

LPT) of engines. Figure 2.5 shows the main elements of the engine model used in C-MAPSS. The dataset used in this case study was simulated based on the assumption that the degradation of engines resulted from wear and tear of one single component (i.e., high pressure chamber, HPC) of the engines based on the usage pattern, under constant operating condition [11] . Also, in this dataset, the damage accumulation during a particular flight cannot be calculated directly based on flight duration and flight conditions, and we have to rely on the degradation signals recorded by multiple sensors during or right after each flight.

Table 2.3: 21 outputs for degradation modeling

Index	Symbol	Description	units
1	T2	Total temperature at fan inlet	°R
2	T24	Total temperature at LPC outlet	°R
3	T30	Total temperature at HPC outlet	°R
4	T50	Total temperature at LPT outlet	°R
5	P2	Pressure at fan inlet	psia
6	P15	Total pressure in bypass-duct	psia
7	P30	Total pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	epr	Engine pressure ratio (P50/P2)	–
11	Ps30	Static pressure at HPC outlet	psia
12	phi	Ratio of fuel flow to Ps30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass Ratio	–
16	farB	Burner fuel-air ratio	–
17	htBleed	Bleed Enthalpy	–
18	Nf_dmd	Demanded fan speed	rpm
19	PCNfR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

We applied our prognostic methodology by first performing sensor selection using four different (log)-location-scale distributions, i.e., normal, lognormal, SEV and Weibull distributions. Similar to the procedure presented in the simulation study (Section 2.5), we

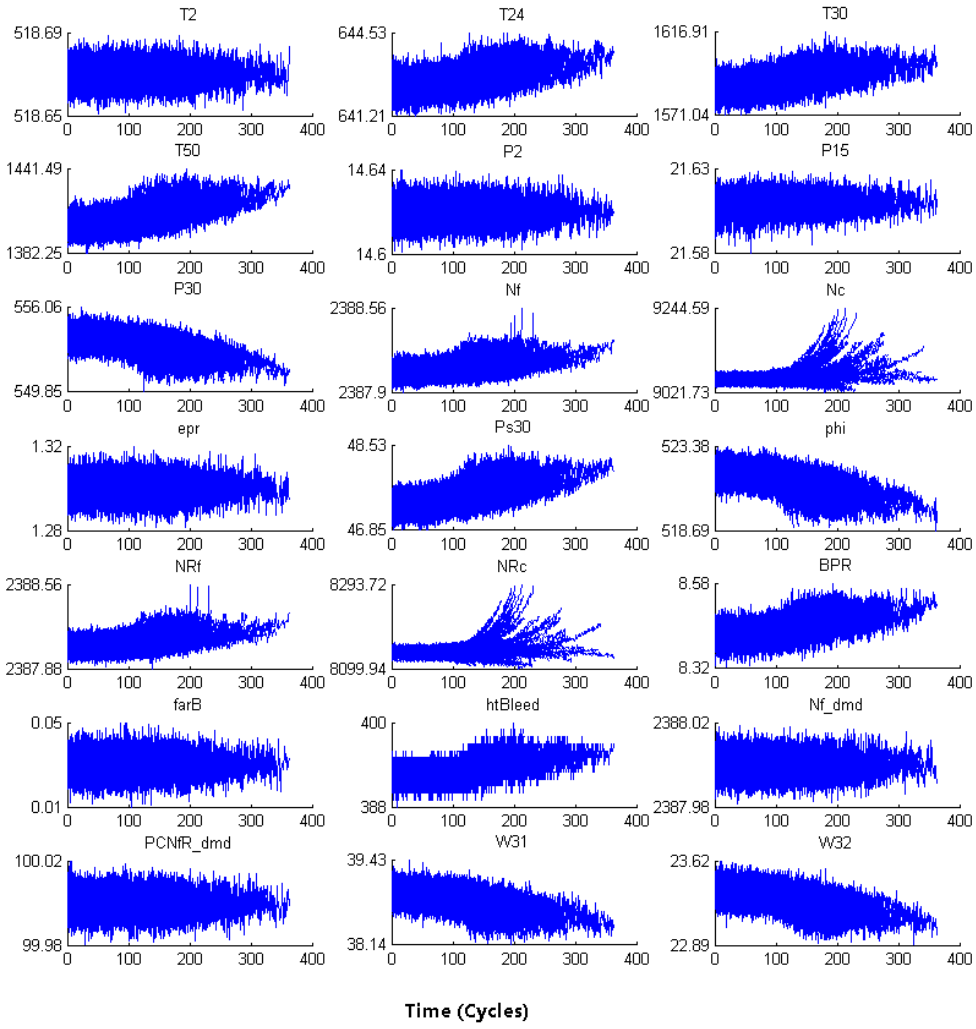


Figure 2.6: Degradation signals from 21 sensors of 100 training aircraft turbofan engines.

created 65 subsamples with different time domains for sensor selection. The proposed sensor selection approach is then applied to these 65 subsamples and the selection rate of each sensors is computed and reported in Table 2.4. Sensors with the selection rate higher than 50% are selected as informative sensors. As can be seen from Table 2.4, sensors 4, 15, 17, 20, were selected by normal and lognormal distributions and sensor 4, 17, 20 were selected by SEV and Weibull distribution.

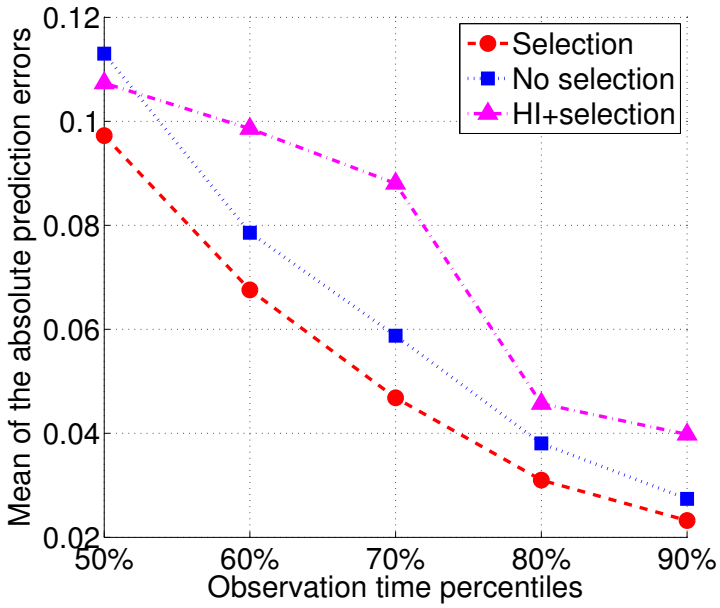
To select the most suitable model, we calculate the corresponding AIC for these dis-

tributions and select the one with the lowest AIC. The calculated AIC values for normal, lognormal, SEV and Weibull are 989.9, 977.7, 1027.2 and 1006.8, respectively, suggesting that the lognormal distribution is likely the most appropriate. Subsequently, the PD-signals of the selected sensors are then fused using Multivariate FPCA and the fused features are used to develop the subsequent prognostic model based on lognormal regression.

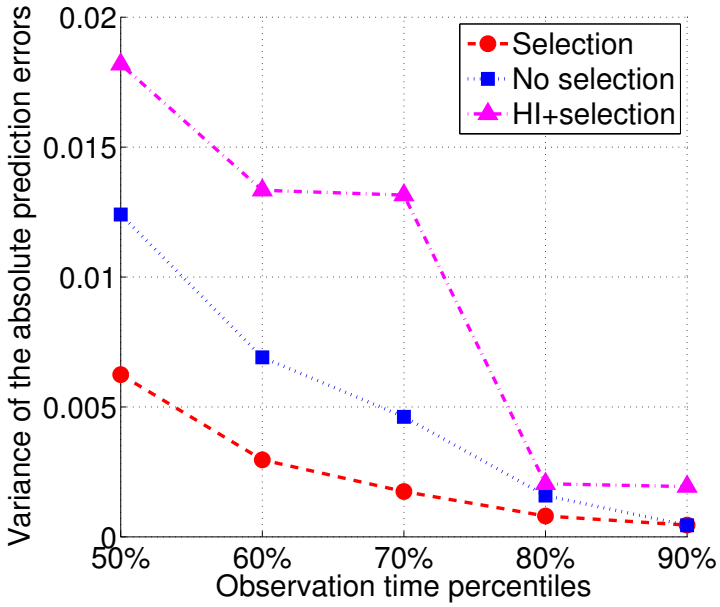
Table 2.4: Sensor selection results for aircraft engine dataset.

Sensor	Normal	Lognormal	SEV	Weibull
1	0%	0%	0%	0%
2	35.0%	45.0%	15.0%	30.0%
3	22.5%	30.0%	17.5%	22.5%
4	**55.0%**	**55.0%**	**62.5%**	**65.0%**
5	0%	0%	0%	0%
6	0%	0%	0%	0%
7	12.5%	7.5%	17.5%	15.0%
8	0%	0%	0%	0%
9	0%	0%	0%	0%
10	0%	0%	0%	0%
11	35.0%	20.0%	45.0%	45.0%
12	0%	0%	0%	0%
13	0%	0%	0%	0%
14	0%	0%	0%	0%
15	**67.5**%	**70.0**%	27.5%	42.5%
16	0%	0%	0%	0%
17	**82.5**%	**85.0**%	**55.0%**	**65.0%**
18	0%	0%	0%	0%
19	0%	0%	0%	0%
20	**77.5**%	**90.0**%	**55.0%**	**67.5%**
21	40.0%	35.0%	47.5%	45.0%

The performance of our model is compared with two benchmarks in terms of prediction accuracy and precision. The first benchmark, "no selection", is the same benchmark used in the simulation study in which all the sensors were used for prediction. The second benchmark is the health index methodology ("HI+selection") proposed by [32] in which the authors used the same aircraft dataset. In [32], the authors visually selected 11 sensors



(a) Mean



(b) Variance

Figure 2.7: Residual life prediction errors for multi-sensor aircraft turbofan engines.

out of the total 21 sensors, and created a composite degradation index by fusing these 11 individual sensor signals. Since all degradation signals include a stationary phase with no degradation at the beginning period [11], we calculate the prediction errors at only the 50%,

$60\%, \ldots, 90\%$ of lifetime in a similar manner to the simulation study.

The mean and variance of the absolute prediction errors are summarized in Figure 2.7. Comparing the mean and variance of the absolute prediction errors of our method with those of benchmarks indicates the overall superiority of our model and as expected the importance of the sensor selection. The mean prediction errors of "selection" is lower than those of two other benchmarks at all life percentiles. In addition, our method has signifficantly smaller variance than the health index method, especially at earlier life percentiles, which again supports the benefits of sensor selection.

2.7 Conclusion

Prognostic of complex systems monitored by multiple sensors is an important yet challenging problem. In this chapter, we proposed a multi-sensor prognostic methodology that utilizes multistream signals to predict RULs of partially degraded systems. The proposed methodology consists of three steps. First, a sensor selection procedure was developed based on penalized (log)-location-scale functional regression. Specifically, we used the FPCA to extract the features of degradation signals, and then applied penalized regression to select informative sensors. In the second step, we utilized the Multivariate FPCA technique to fuse the PD-signals while considering the inter-relationship of signal streams. Finally, using the extracted fused features from Multivariate FPCA, an adaptive (log)-location-scale regression model with regularization was built for dynamic prediction of RULs.

We studied and compared the performance of our proposed methodology with different benchmarks using simulations. The results indicated that our methodology outperforms the benchmarks in terms of both the mean and variance of prediction errors. Moreover, simulation results showed that the proposed sensor selection procedure can effectively remove non-informative sensors, which in turn resulted in a more accurate and precise predictions. We also validated the effectiveness of our methodology using a simulated aircraft turbofan

engine dataset from NASA repository. The results indicated that the sensor selection algorithm in our model can improve both the prediction accuracy and precision. The model developed in this chapter only focuses on the multi-sensor systems with single failure mode. Development of a multistream prognostic methodology for systems with multiple failure modes is an important topic for future research.

CHAPTER 3
SCALABLE PROGNOSTIC MODELS FOR LARGE-SCALE CONDITION MONITORING APPLICATIONS

3.1 Introduction

Many capital-intensive engineering systems are monitored by hundreds (and sometimes thousands) of sensors. For example, a typical gas turbine is equipped with over $2,000$ sensors that are used to monitor vibrations, temperatures, and pressures related to the condition and performance of many of its components [51]. Condition monitoring is the process of using sensor signals to detect faults. In cases where the sensor signals possess trends that are strongly correlated with the progression of physical degradation, they can be very useful for prognostic purposes. In prognostics, degradation-based signal trends are modeled to predict remaining lifetime and operational risk. There is a plethora of prognostic models in the literature. Commonly used techniques include random coefficients models, Brownian motion, gamma process, and Markov chains all of which have shown great promise in modeling degradation signals to predict lifetime and/or remaining lifetime [1, 52, 8, 7, 13, 4, 5, 2]. Typically, a degradation signal is computed from specific features obtained from the raw sensor data. A large portion of the degradation modeling literature assumes that degradation signals originate from single-sensor applications. By doing so they often circumvent the underlying challenges that arise in multi-sensor applications, i.e., how to combine degradation signals from different sensors. Although one can argue that single-sensor prognostic models can still be used in multi-sensor settings by computing a single "aggregate" degradation signal, such an approach is not trivial and can create additional challenges in of its own.

Multi-sensor application often involve complex equipment that typically undergo mul-

tifaceted degradation processes. Using multiple sensors potentially captures different aspects of complicated degradation processes that sometimes exhibit different failure modes. Overall, the data in such cases is much richer and can lead to more accurate failure predictions. In [53], the authors argue that the partial information obtained from different sensors has the potential of providing more accurate diagnostic and prognostic capabilities. Interesting challenges arise when we consider the prognostics problem within a multi-sensor setting. One of the key aspects in this setting is how to systematically combine information from multiple sensors from the same equipment, otherwise known as fusion. [53] provides a review of multisensor data fusion approaches and classifies the techniques based on the level at which fusion is performed; data, feature, and decision levels. Data-level fusion directly integrates information of the raw data from multiple sensors [54, 55, 56, 57]. Feature-level fusion combines feature information extracted from the raw data [58]. Decision-level fusion focuses on integrating different diagnostic or prognostic results [59, 60]. A large portion of the fusion literature utilizes artificial intelligence approaches, such as neural network and fuzzy logic, however they are mostly focused on fault detection and diagnostics. Other approaches rely on computing some form of aggregate degradation signal that is constructed by taking a weighted combination of various types of degradation signals [32, 33, 34, 24, 22, 61, 62]. For example, in [32, 33, 34] an aggregate degradation signal (health index) was computed by taking a weighted linear combination of different degradation signals. The weights were computed using a specialized optimization algorithm. In [24], PCA was applied at each observation time and the first principal component was used to construct an aggregate degradation signal. In [61], PCA was also applied to failure data with the goal of constructing a feature space. Data at each observation time was then projected onto the feature space, and the euclidean distance was used to infer degradation. Although these models have shown promise in their respective settings, yet almost all have been exclusively validated using small-sized data sets. It is also not clear how these models would perform in Big Data settings involving large scale data, and whether they

would even scale to such settings. Furthermore, the concept of computing and modeling an aggregate degradation signal can be tricky because it is not clear how the resulting signal captures cross-correlations that exists among different degradation signals.

The volume and dimensionality of condition monitoring data generated by today's industrial applications has become prohibitive. For example, some optical sensors used for turbine blade crack detection generate 600 gigabytes per day – almost 7 times Twitter daily volume [18]. This chapter has two key contributions. First, we develop a prognostic modeling framework that can scale with the size of the condition monitoring data. Second, our methodology models the cross-correlation of different degradation signals–a critical aspect that is often not considered by many conventional modeling approaches. If properly modeled, signal correlations often contain valuable information that potentially improves prediction accuracy. Our methodology is based on using tools from functional data analysis to systematically extract and combine features from different degradation signals, and subsequently use these features to predict remaining lifetimes of partially degraded equipment. Specifically, we use FPCA to develop signal fusion algorithms, and functional regression to predict the remaining lifetime. FPCA provides a low-dimensional and parsimonious representation of functional data (degradation signals in our case). It works by first constructing a low-dimensional space that preserves most variations of the degradation signals. Degradation signals are then projected to the space, and signal features known as FPC-scores are extracted. FPC-scores are critical in that they encapsulate unique features of the original degradation signals in the low-dimensional space [48, 63, 64]. To predict remaining lifetimes, we use adaptive functional (log)-location-scale regression to model the relationship between the fused signal features (FPC-scores) and TTF. Functional regression is an extension of ordinary regression that accounts for cases in which predictors are random functions and responses are scalars or functions [17].

Functional principal component analysis is inherently computationally expensive because it involves matrix decomposition, e.g., singular value decomposition and/or eigen

decomposition. In large scale settings involving large amounts of data, this aspect can become a major impediment to the scalability of FPCA. This problem becomes even more prominent in the multi-sensor applications like the ones considered in this chapter. Conventional FPCA is only suited for modeling multiple realizations of the *same type of functional data*. That is, a sample of the same type of degradation signal observed from similar units. As such, it is not well-suited for applications where each unit has multiple types of degradation signals. Multivariate FPCA is an extension of FPCA that allows us to utilize the FPCA framework for multi-sensor applications. It works by concatenating various types of degradation signals, thus the resulting signal matrix becomes even much lager. From another perspective, functional regression, in its classical sense, is used for one-shot estimation of the response variable, TTF in our case. Our goal, however, is to be able to integrate real-time degradation signals observed from fielded equipment to update predictions of remaining lifetime on a constant basis. To achieve this, we exploit an adaptive version of functional regression known as time-varying functional regression [19]. Time-varying functional regression allows us to recalibrate our model based on the unique degradation signals of each unit. However, this process results in a new signal matrix each time. As a result, matrix decomposition needs to be performed repeatedly as new data becomes available from the field.

We address the computational challenges by leveraging recent developments in randomized algorithms used for numerical linear algebra. Specifically, we utilize randomized low-rank approximation (RLA) in key steps within the FPCA methodology. Low-rank approximation focuses on building a matrix with the smallest rank but preserves most of the useful information of the original matrix. Examples of low-rank approximation techniques include singular value decomposition (SVD), pivoted QR factorization and eigen decomposition (ED). Conventional algorithms for computing the aforementioned low-rank approximations include truncated SVD [65], Golub-Businger algorithm [66], Gu and Eisenstat's strong rank-revealing QR [67] and Lanczos method [68]. However, these methods are still

computationally expensive. Randomized methods like the RLA were specifically designed to overcome computational challenges involved in various matrix factorization and decomposition operations such as SVD and ED. RLA works by first computing an approximation to the range (also known as column space) of a matrix via randomized sampling. In our case the matrix of concern is the signal matrix. The signal matrix is then projected to the approximated range, and a factorization of the resulting low-rank matrix is computed. Although RLA is an approximation technique, its error bounds have been well-studied [20]. One of the key contributions of this chapter is that we enhance the scalability of (multivariate) FPCA by exploiting RLA. However, this integration is not trivial. For example, a key aspect in RLA is that is requires that the rank (number of principal components) of the matrix be known in advance, which is not the case in our framework. Details of the integration are discussed in Section 3.5.

The remainder of the chapter is organized as follows. Section 6.2 presents the prognostic modeling framework and Section 3.3 introduces the parameter estimation. Section 3.4 presents the real-time updating strategy, and Section 3.5 discusses the computational challenges and presents approaches to addressing them. Sections 7.5 and 7.7 evaluate the performance of the model via simulation and case studies, respectively. Finally, Section 7.8 concludes.

3.2 Scalable prognostic modeling framework

The basis of our approach is to treat degradation signals from each sensor as noisy realizations of a stochastic process, and then regress them against the TTFs using a functional (log)-location-scale regression model. We assume that raw sensor signals can be synthesized into (numerical) degradation signals, and that each sensor has its corresponding degradation signal. Our framework is based on a nonparametric modeling approach that consists of two key steps, (1) signal fusion, and (2) prognostics. Nonparametric methods are relatively flexible and can be used to model complex trends. They are also general

enough to encompass simpler parametric forms. The first step of our methodology develops two alternative kinds of signal fusion algorithms; *Multivariate* FPCA and *Hierarchical* FPCA (see Figure 3.1). *Multivariate* FPCA is an extension of FPCA and works by concatenating different types of degradation signals into a single vector. FPCA is then applied to the concatenated vector in a conventional manner to extract "fused signal features", i.e., FPC-scores of the concatenated signals. The second approach, *Hierarchical* FPCA, works by first applying FPCA to the individual degradation signals (grouped by sensor type), and then extracting their corresponding FPC-scores. (Recall, FPCA is used to model variations in the same type of functional data, i.e., degradation signals computed from the same sensor on different units.) Next, *Hierarchical* FPCA concatenates these FPC-scores and computes a set of fused signal features by applying regular PCA on the concatenated vector (contrast this with the *Multivariate* FPCA fusion method where concatenation is done on the degradation signals). Both *Multivariate* and *Hierarchical* FPCA provide a systematic methodology for performing sensor fusion that is unique in two key aspects. First, it models the cross-correlations of different types of degradation signals. This provides much richer insights into complex degradation processes. Secondly, their computational efficiency can be significantly enhanced (as will be shown later), thus enabling their scalability to industrial Big Data settings.

Prognostics is the second step of our methodology. In prognostics, we use an adaptive functional LLS regression model to estimate and continuously update remaining useful life of fielded systems. Functional regression is a regression model where the predictor is random functions and the response can be either random functions or scalars. In its simplest form, by incorporating FPCA, functional regression can be used to model the FPC-scores as the predictor and lifetime as the response variable. However, this setting does not allow the integration of real-time signals observed from fielded equipment. To address this, we focus on an adaptive version of functional regression known as time-varying functional regression. Time-varying functional regression allows us to leverage real-time degrada-

tion signals observed from individual equipment operating in the field in order to revise predictions of remaining lifetime based on the latest degradation characteristics of each equipment.

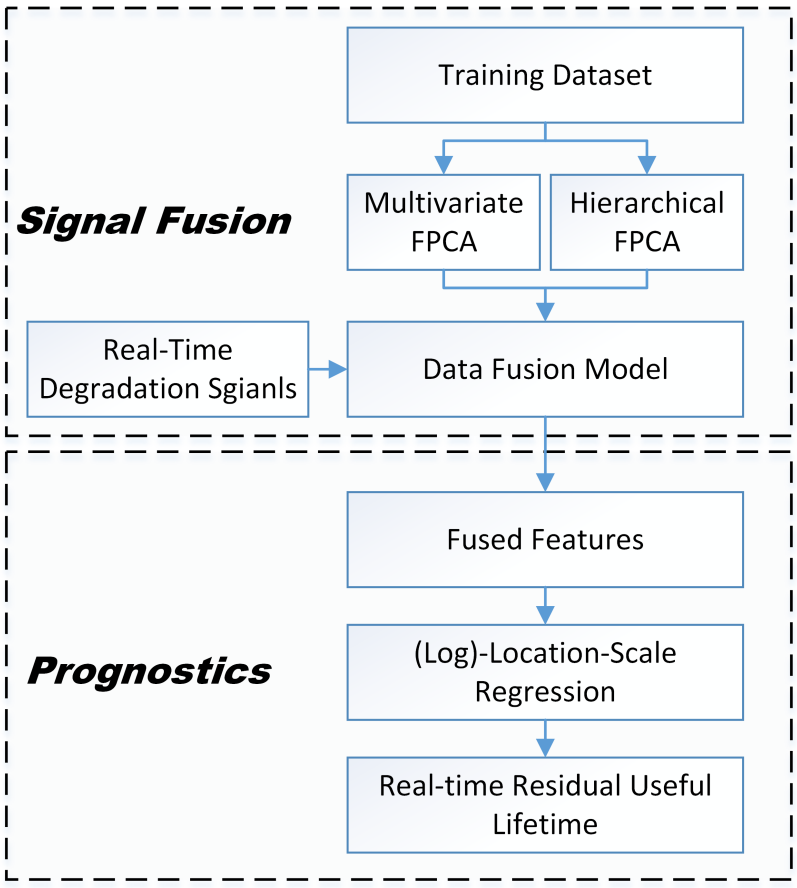


Figure 3.1: The framework of our methodology.

We consider a settings where multiple units of a fleet of equipment are monitored in real-time. Such a setting is not unrealistic. Many companies in the energy, airlines and railway sectors have centers specialized for monitoring turbines, aircraft engines and locomotives. We assume that these assets are equipped with multiple sensors and that data from each sensor is synthesized into one type of degradation signal. We also assume that there exists a database of historical degradation signals from a set of identical equipment along with their corresponding TTFs (training set). Let us consider a training dataset of N units (we use unit to refers to an asset, equipment, or machine). Each unit is monitored by

P sensors. In other words, we have N units with N TTFs, and P degradation signals for each unit. Without loss of generality, we assume that each sensor corresponds to only one degradation signal. We would like to point out that, in reality, it may be possible that more than one degradation signal can be synthesized from the data measured by a single sensor. However, this will not affect the application of our models. Let $s_{i,p}(t)$ denote the degradation signal obtained from sensor p of unit i, where $i = 1, \ldots, N$ and $p = 1, \ldots, P$. We assume that $s_{i,p}(t)$ are independent noisy realizations of a smooth random function, $s_p(\cdot)$, in a bounded time domain [0,T], with an unknown mean function, $\mu_p(t)$, and a covariance function, $C_p(t, t')$. Here, $\mu(t)$ represents the general trend followed by the degradation signals and is deterministic. $C_p(t, t')$ models the deviations from that trend that are caused by unit-to-unit variability in the degradation process and the inherent signal noise. Based on the premise that the TTF of unit i, $\tilde{Y}_i$, can be predicted by its degradation signals, $\{s_{i,p}(t)\}_{p=1}^{P}$, we establish the following (log)-location-scale regression model:

$$Pr(Y_i \leq y) = \Omega\left(\frac{y - \pi(s_{i,p}(t))}{\sigma}\right), \tag{3.1}$$

where $Y_i = \tilde{Y}_i$ if $\tilde{Y}_i$ follows a location-scale distribution, and $Y_i = \ln(\tilde{Y}_i)$ if $\tilde{Y}_i$ follows a log-location-scale distribution. Ω is the cumulative distribution function of the location-scale distribution. For example, $\Omega(z) = \int_{-\infty}^{z}(1/\sqrt{2\pi}\exp(-x^2/2))dx$ for the case of a normal distribution, $\Omega(z) = 1 - \exp(-\exp(z))$ for a smallest extreme value distribution, and $\Omega(z) = \exp(z)/(1 + \exp(z))$ for a logistic distribution. σ is the scale parameter and is assumed to be fixed across units; and $\pi(s_{i,p}(t))$ is the location parameter and is assumed to be a function of the degradation signals. The functional form of $\pi(s_{i,p}(t))$ will depend heavily on the fusion method being. *Multivariate* FPCA defines the location parameter by $\pi(s_{i,p}(t)) = \alpha_0 + \int_0^T \boldsymbol{\alpha}(t)^\top \boldsymbol{s}_i(t)dt$, where $\boldsymbol{s}_i(t) = (s_{i,1}(t), \ldots, s_{i,P}(t))^\top$ is the signal vector, α_0 is the intercept and $\boldsymbol{\alpha}(t) = (\alpha_1(t), \ldots, \alpha_P(t))^\top$ is the vector of the coefficients of the regression function. In *Hierarchical* FPCA, the location parameter is expressed as $\pi(s_{i,p}(t)) = \theta_0 + \sum_{m=1}^{M}\theta_m \upsilon_{i,m}$, where $\upsilon_{i,m}$ represents the FPC-scores extracted by using

Hierarchical FPCA and θ_0, θ_m are the regression coefficients that can be estimated using the training data.

Our framework considers a relatively rich analytics environment in which degradation signals are high-dimensional and where signals from the same unit exhibit varying levels of cross-correlation that need to be leveraged in order to enhance failure predictability. This is why we focus on signal fusion techniques that reduce the dimensionality of the data and simultaneously compute fused signal features that attempt to capture and model these complex relationships. Both *Multivariate* and *Hierarchical* FPCA provide the means to achieve this objective. The next section covers the details of the two signal fusion methodologies.

3.2.1 Multi-sensor signal fusion using Multivariate FPCA

Multivariate FPCA is an extension of FPCA and works by concatenating different types of degradation signals into a single vector. FPCA is then applied to the concatenated vector in a conventional manner to extract fused signal features. Let $\boldsymbol{s}_i(t) = (s_{i,1}(t), \ldots, s_{i,P}(t))^\top, t \in [0, T]$ represent the concatenated signal with mean function $\boldsymbol{\mu}(t) = (\mu_1(t), \ldots, \mu_P(t))^\top$; and a covariance function that is given by the following $P \times P$ block matrix:

$$\boldsymbol{C}(t, t') = \begin{pmatrix} C_{1,1}(t, t') & \ldots & C_{1,P}(t, t') \\ \vdots & \ddots & \vdots \\ C_{P,1}(t, t') & \ldots & C_{P,P}(t, t') \end{pmatrix}, \tag{3.2}$$

where $C_{g,h}(t, t') = Cov(s_g(t), s_h(t'))$, for $g = 1, \ldots, P$ and $h = 1, \ldots, P$, with $t, t' \in [0, T]$.

Using Mercer's theorem [36], $\boldsymbol{C}(t, t')$ can be decomposed as,

$$\boldsymbol{C}(t, t') = \sum_{k=1}^{\infty} \eta_k \boldsymbol{\psi}_k(t) \boldsymbol{\psi}_k(t')^\top, \ t, t' \in [0, T] \tag{3.3}$$

where $\eta_1 \geq \eta_2 \geq \ldots$, are ordered nonnegative eigenvalues, and $\boldsymbol{\psi}_k(t) = (\psi_{k,1}(t), \ldots, \psi_{k,P}(t))^\top$ for $k = 1, 2, \ldots$ are the corresponding eigenfunctions . Using this decomposition, we can

rewrite $\boldsymbol{s}_i(t)$ as follows:

$$\boldsymbol{s}_i(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{\infty} \zeta_{i,k}\boldsymbol{\psi}_k(t), \tag{3.4}$$

where $\zeta_{i,k} = \int_0^T (\boldsymbol{s}_i(t) - \boldsymbol{\mu}(t))^\top \boldsymbol{\psi}_k(t)dt$ represent the FPC-scores, which are independent random variables with mean 0 and variance η_k. It is often sufficient to use a few eigenfunctions corresponding to the largest eigenvalues to approximate signals with a reasonable accuracy. Using only K eigenfunctions, equation (3.4) can now be rewritten as,

$$\boldsymbol{s}_i(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{K} \zeta_{i,k}\boldsymbol{\psi}_k(t). \tag{3.5}$$

Recall that $\pi(s_{i,p}(t)) = \alpha_0 + \int_0^T \boldsymbol{\alpha}(t)^\top \boldsymbol{s}_i(t)dt$. Since the set of eigenfunctions $\boldsymbol{\psi}_k(t)$ for $k = 1, 2, \ldots$ forms a complete orthonormal basis, $\boldsymbol{\alpha}(t)$ can be expanded to $\boldsymbol{\alpha}(t) = \sum_{k=1}^{\infty} \beta_k \boldsymbol{\psi}_k(t)$. Therefore, the location parameter can be expressed as follows (details of the derivation can be found in Appendix B):

$$\pi(s_{i,p}(t)) \approx \beta_0 + \sum_{k=1}^{K} \beta_k \zeta_{i,k}. \tag{3.6}$$

Using *Multivariate* FPCA, the (log)-location-scale regression model in Equation (3.1) can be written as,

$$Pr(Y_i \leq y) = \Omega\left(\frac{y - \beta_0 - \sum_{k=1}^{K} \beta_k \zeta_{i,k}}{\sigma}\right). \tag{3.7}$$

We can model different failure distributions. For example, for failure times that follow a normal/lognormal distribution $Pr(Y_i \leq y) = \int_{-\infty}^{\frac{y-\beta_0-\sum_{k=1}^{K}\beta_k\zeta_{i,k}}{\sigma}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})dx$; for a smallest extreme value/Weibull distribution, $Pr(Y_i \leq y) = 1 - \exp(-\exp(\frac{y-\beta_0-\sum_{k=1}^{K}\beta_k\zeta_{i,k}}{\sigma}))$.

3.2.2 Multi-sensor signal fusion using Hierarchical FPCA

Hierarchical FPCA works by first applying FPCA to the individual degradation signals (grouped by sensor type), and then extracting their corresponding FPC-scores. Next, it concatenates these FPC-scores and computes a set of fused signal features by applying regular PCA on the concatenated vector. In other words, using Mercer's theorem, we can express the covariance matrix $C_p(t,t')$ of the degradation signals of sensor p, $s_{i,p}(t)$, as follows:

$$C_p(t,t') = \sum_{k=1}^{\infty} \lambda_{k,p}\phi_{k,p}(t)\phi_{k,p}(t'), \tag{3.8}$$

where $\phi_{k,p}(t)$ for $k = 1, 2, \ldots$ represent the orthogonal eigenfunctions and $\lambda_{1,p} \geq \lambda_{2,p} \geq \ldots$ the ordered nonnegative eigenvalues. Thus, the degradation signals obtained from sensor p can now be expressed can be approximated by the model below in a similar spirit to the model (3.5) discussed in the previous section.

$$s_{i,p}(t) \approx \mu_p(t) + \sum_{k=1}^{K_p} \xi_{i,k,p}\phi_{k,p}(t), \tag{3.9}$$

where $\xi_{i,k,p}$ are the FPC-scores.

To capture the cross-correlation among degradation signals from different sensors, the individual FPC-scores associated with each sensor type are first concatenated in the following matrix $\mathbf{\Xi}$,

$$\mathbf{\Xi} = \begin{bmatrix} \overbrace{\xi_{1,1,1} \cdots \xi_{1,K_1,1}}^{\text{sensor 1}} & \overbrace{\xi_{1,1,2} \cdots \xi_{1,K_2,2}}^{\text{sensor 2}} & \cdots & \overbrace{\xi_{1,1,P} \cdots \xi_{1,K_P,P}}^{\text{sensor P}} \\ \vdots \ddots \vdots & \vdots \ddots \vdots & \ldots & \vdots \ddots \vdots \\ \xi_{N,1,1} \cdots \xi_{N,K_1,1} & \xi_{N,1,2} \cdots \xi_{N,K_2,2} & \cdots & \xi_{N,1,P} \cdots \xi_{N,K_P,P} \end{bmatrix}_{N \times \sum_{p=1}^{P} K_p}.$$

Next, regular PCA is then applied to $\mathbf{\Xi}$. Singular value decomposition is used to de-

compose matrix $\mathbf{\Xi}$ and obtain the eigenvalues and eigenvectors of covariance matrix of $\mathbf{\Xi}$. Using the eigenvectors corresponding to the first M largest eigenvalues, denoted by $\{\boldsymbol{u}_m\}$, we calculate fused features as,

$$\upsilon_{i,m} = \tilde{\mathbf{\Xi}}(i,:)\boldsymbol{u}_m, \tag{3.10}$$

where $\tilde{\mathbf{\Xi}}$ is the centered matrix $\mathbf{\Xi}$ and $\upsilon_{i,m}$ is the mth FPC-score of unit i. Hence the location parameter can be expressed as,

$$\pi_i(s_{i,p}(t)) = \theta_0 + \sum_{m=1}^{M} \theta_m \upsilon_{i,m}, \tag{3.11}$$

where θ_0 and θ_m are the regression coefficients.

Using *Hierarchical* approach, the (log)-location-scale regression model in (3.1) can now be expressed in the following form,

$$Pr(Y_i \leq y) = \Omega\left(\frac{y - \theta_0 - \sum_{m=1}^{M} \theta_m \upsilon_{i,m}}{\sigma}\right). \tag{3.12}$$

Similar to the previous methodology, we can also express different forms of failure time distributions.

3.3 Parameter estimation

This framework assumes that there exits a training data set that can be used for estimating our multi-sensor prognostic model. We begin by disussing the estimation of the eigenvectors and fused features (i.e., $\zeta_{i,k}$ and $\upsilon_{i,m}$) using the "training" degradation signals. The fused features along with the TTFs in the training dataset are then used to estimate the location and scale parameters of the lifetime distribution using maximum likelihood estimation (MLE). Real-time updating of this distribution will be explained in section 3.4.

3.3.1 Estimating fused signal features

To facilitate the estimation process, we express the degradation signal from sensor p of unit i in its discrete form as $s_{i,p}(t_j)$, where t_j is the discrete observation time point, $j = 1, \ldots, J$, and J is the total observation number.

For the *Multivariate*FPCA case, SVD is applied to a matrix of centered concatenated degradation signals $\{\boldsymbol{s}_i(t_j) - \hat{\boldsymbol{\mu}}(t_j)\}$, $j = 1, \ldots, J$. The K right singular vectors corresponding to the K largest singular values produce the estimated eigenfunctions, $\hat{\boldsymbol{\psi}}_k(t_j)$, $k = 1, \ldots, K$. K is determined by using the fraction variance explained (FVE) criterion, which will be discussed in greater detail in Section 3.5. The fused signal features, $\zeta_{i,k} = \int_0^T (\boldsymbol{s}_i(t) - \boldsymbol{\mu}(t))^\top \boldsymbol{\psi}_k(t)dt$, are estimated numerically as follows: $\hat{\zeta}_{i,k} = \sum_{j=1}^{J} (\boldsymbol{s}_i(t_j) - \hat{\boldsymbol{\mu}}(t_j))^\top \hat{\boldsymbol{\psi}}_k(t_j)(t_j - t_{j-1})$, where $t_0 = 0$.

For the *Hierarchical* FPCA signal fusion method, SVD is applied on the centered degradation signals from each sensor $\{s_{i,p}(t_j) - \hat{\mu}_p(t_j)\}$, separately. The resulting K_p right singular vectors corresponding to the K_p largest singular values are the estimated eigenfunctions associated with each degradation signal type (or each sensor). That is, $\hat{\phi}_{k,p}(t_j), k = 1, \ldots, K_p$. Similarly, we determine the constant K_p from the FVE criterion and calculate the FPC-scores for the sensor p using numerical integration. That is, $\hat{\xi}_{i,k,p} = \sum_{j=1}^{J} (s_{i,p}(t_j) - \hat{\mu}_p(t_j))\hat{\phi}_{k,p}(t_j)(t_j - t_{j-1})$, where $t_0 = 0$. Next, the estimated FPC-scores are concatenated to form the matrix $\hat{\boldsymbol{\Xi}}$, which is decomposed using SVD to find the eigenvectors. The eigenvectors corresponding to the first M largest eigenvalues are used to calculate the fused signal features, $\hat{v}_{i,m}$, defined earlier by Equation (3.10) of Section 3.2.2, and where $m = 1, \ldots, M$. M is again determined using the FVE criterion.

3.3.2 Estimating location-scale regression parameters

Maximum likelihood estimation is used to estimate the location and scale parameters of the regression model expressed in (3.1). The likelihood function is specifically expressed

as follows:

$$L(\boldsymbol{\alpha}) = \prod_{i=1}^{N} \frac{1}{\sigma} \omega \left(\frac{Y_i - \pi(s_{i,p}(t))}{\sigma} \right), \tag{3.13}$$

where $\pi(s_{i,p}(t)) = \beta_0 + \sum_{k=1}^{K} \beta_k \zeta_{i,k}$, and $\boldsymbol{\alpha} = (\beta_0, \beta_k, \sigma)$ is the vector of unknown parameters for the case of the *Multivariate* FPCA methodology, and $\pi(s_{i,p}(t)) = \theta_0 + \sum_{m=1}^{M} \theta_m \upsilon_{i,m}$, and $\boldsymbol{\alpha} = (\theta_0, \theta_m, \sigma)$, for the *Hierarchical* case. $Y_i = \tilde{Y}_i$ for location-scale distributions whereas $Y_i = \ln(\tilde{Y}_i)$ for log-location-scale distributions, where Y_i is the TTF of unit i. σ is the scale parameter, and ω is the probability density function of the designated location-scale distribution. Using the following reparametrization; $\tilde{\sigma} = 1/\sigma, \tilde{\beta}_0 = \beta_0/\sigma, \tilde{\beta}_k = \beta_k/\sigma$, the log-likelihood function can be rewritten as follows:

$$\ell(\tilde{\boldsymbol{\alpha}}) = N \log \tilde{\sigma} + \sum_{i=1}^{N} \log \omega \left(Y_i \tilde{\sigma} - \tilde{\pi}(s_{i,p}(t)) \right). \tag{3.14}$$

where $\tilde{\pi}(s_{i,p}(t))$ and $\tilde{\boldsymbol{\alpha}}$ are defined for the *Multivariate* and *Hierarchical* cases in a similar fashion to the above, using the reparametrized terms. Note that, $Y_i \tilde{\sigma} - \tilde{\pi}(s_{i,p}(t))$ of Equation (3.14) is concave, and hence its logarithm is also concave. Thus, the likelihood function, $\ell(\boldsymbol{\alpha})$, will be concave if the function $\ln \omega(\cdot)$ is also concave. In other words, if the pdf $\omega(\cdot)$ is log-concave, the log-likelihood function in Equation (3.14) will be concave. As it turns out, the density function for most of the (log)-location-scale distributions (e.g., *normal*, *logistic*, *smallest extreme value*, *generalized log-gamma*, and *log-inverse Gaussian*) are log-concave [69]. For (log)-location-scale distributions with density functions that are not log-concave, we can usually transform them to log-concave distributions. For example, *lognormal*, *log-logistic* and*Weibull* distribution can be easily transformed to *normal*, *logistic* and *smallest extreme value* distribution by taking their logarithm. Equation (3.14) is therefore concave, which implies that MLE will provide a global maximum.

Maximum likelihood estimation computes the expected TTF as a function of the estimated parameters vector $\hat{\boldsymbol{\alpha}}$. For example, mean TTF for the normal distribution is $\hat{\beta}_0 +$

$\sum_{k=1}^{K} \hat{\beta}_k \hat{\zeta}_{i,k}$ and $\hat{\theta}_0 + \sum_{m=1}^{M} \hat{\theta}_m \hat{\upsilon}_{i,m}$ for *Multivariate* and *Hierarchical* FPCA, respectively. Similar estimates for the lognormal distribution are $\exp(\hat{\beta}_0 + \sum_{k=1}^{K} \hat{\beta}_k \hat{\zeta}_{i,k} + \hat{\sigma}^2/2)$ and $\exp(\hat{\theta}_0 + \sum_{m=1}^{M} \hat{\theta}_m \hat{\upsilon}_{i,m} + \hat{\sigma}^2/2)$, and for the Weibull, $\exp(\hat{\beta}_0 + \sum_{k=1}^{K} \hat{\beta}_k \hat{\zeta}_{i,k})\Gamma(1+\hat{\sigma})$ and $\exp(\hat{\theta}_0 + \sum_{m=1}^{M} \hat{\theta}_m \hat{\upsilon}_{i,m})\Gamma(1+\hat{\sigma})$. The above estimation procedure provides a general approach for model estimation using the training dataset. Our focus, however, is on the ability to leverage real-time degradation signals observed from equipment that are still functioning in the field to estimate the remaining lifetime. The in-situ signals capture the latest degradation states of each unit, and hence, can provide significant insights into their remaining lifetimes.

3.4 Real-time predictions of RUL

The degradation signals observed from fielded units ("observed signal" for short) are used to provide updated estimates of the fused signal features (i.e., $\zeta_{i,k}$ and $\upsilon_{i,m}$). The update fused signal features are then input to the (log)-location-scale regression model to predict revised remaining useful life. The overall approach rests on the idea of combining the observed signals with the training signals to compute the updated signal features either by using the *Multivariate* or the *Hierarchical* methodologies. However, since both methods employ a nonparametric approach, the signals must share the same time domain. In other words, when combining an observed degradation signal from a fielded unit with degradation signals of units in the training dataset, all the signals for a given sensor must share the same time domain. Since the TTFs of different units are not the same, the lengths of their degradation signals are therefore going to be different. This issue is resolved by using an adaptive approach to update the fused signal features [49]. The idea of this approach is that, training units with lifetimes shorter than the current observation time are are excluded from the subset of signals used to re-estimate the fused features. Thus, only the training units whose lifetime is longer than the current observation time of the observed degradation signals are chosen for updating the signal features. By default, the chosen training degra-

dation signals extend beyond the latest observation epoch. Thus, by truncating them at the latest observation ecpoh, all of the signals (training and observed) now share the same time domain. This new subset of data is the basis for computing the updated fused signal features using *Multivariate* or *Hierarchical* FPCA. Figure 2.2 illustrated how the adaptive updating approach works. In each graph, the dotted lines represent the entire set of training signals. The continuous part marks the part of the data used for the estimation process. The signal marked with the thick continuous line represents the portion of the degradation signal of the test unit that has been observed up to time t^*.

We summarize the process of predicting and updating the RUL of a fielded equipment as follows: Each time new degradation signals are observed, training degradation signals that satisfy the criterion mentioned earlier are selected and truncated at the observed time epoch. Next, sensor fusion is performed on the truncated training signals using either the *Multivariate* or the *Hierarchical* FPCA methodologies discussed earlier. The resulting fused signal features are then used to build a (log)-location-scale regression model where the TTFs (of the truncated training signals) are the dependent variables. The regression parameters are re-estimated using the procedure described in Section 3.3.2). The updated regression model is then used to update the RUL of the unit being monitored. To do this, the observed degradation signals are first projected onto the feature space of the truncated training signals, and their FPC-scores are evaluated. These scores are then input to the updated regression model to estimate the remaining failure time of the unit that was being monitored.

3.5 Computational challenges

Functional principal component analysis is inherently computationally expensive because it involves matrix decomposition, e.g., singular value decomposition (SVD) and/or eigen decomposition (ED). In large scale settings involving multi-sensor applications and large amounts of data, the computations become very time consuming. Additional challenges are

also generated by the updating procedure, which requires repetitive SVD. To improve the computational efficiency of our framework, we use randomized low-rank approximation to significantly speed up SVD decomposition.

Let $\boldsymbol{S}$ denote the matrix of degradation signals. SVD decomposition of $\boldsymbol{S}$ is given by

$$\boldsymbol{S}_{N\times L} = \boldsymbol{U}_{N\times N}\boldsymbol{\Sigma}_{N\times L}\boldsymbol{V}^{\top}_{L\times L}, \tag{3.15}$$

where N is the total number of units, L is the degradation signal length for each unit, and $\boldsymbol{U}$ and $\boldsymbol{V}$ consist of the orthonormal columns that contain the left and right singular vectors of $\boldsymbol{S}$, respectively. Note that the columns of $\boldsymbol{V}$ are equivalent to the eigenvectors of the covariance matrix of $\boldsymbol{S}$. The matrix $\boldsymbol{\Sigma}$ is diagonal and contains singular values $\boldsymbol{\Sigma} = diag(\sigma_1(\boldsymbol{S}), \sigma_2(\boldsymbol{S}), \ldots, \sigma_{\min\{N,L\}}(\boldsymbol{S}))$ where $\sigma_i(\boldsymbol{S})$ is the square root of the eigenvalues, and $\sigma_1(\boldsymbol{S}) \geq \sigma_2(\boldsymbol{S}) \geq \ldots \geq \sigma_{\min\{N,L\}}(\boldsymbol{S})$. In FPCA, the first K largest singular values and their corresponding singular vectors are often selected to approximate the matrix thus,

$$\boldsymbol{S}_{N\times L} \approx \boldsymbol{S}_{(K)} = \boldsymbol{U}_{N\times K}\boldsymbol{\Sigma}_{K\times K}\boldsymbol{V}^{\top}_{K\times L}, \tag{3.16}$$

where $\boldsymbol{S}_{(K)}$ represents the approximated matrix $\boldsymbol{S}$ using its first K singular values and their corresponding singular vectors. Since only the first K singular values and their corresponding singular vectors are important, partial decomposition techniques such as the Golub-Businger algorithm [66], Gu and Eisenstat's strong rank-revealing QR [67], and Lanczos method [68] can be utilized to accelerate the decomposition. However, partial decomposition methods are sometimes unstable and tend to be slow in settings with large scale data. The randomized low-rank approximation algorithm presented by [20] has been shown to be relatively stable and well-suited for large-scale data applications.

3.5.1 Randomized low-rank approximation algorithm

The RLA algorithm consists of two steps as shown in Table 3.1. Step 1 extracts a basis for the range (column space) of a given matrix (the degradation signal matrix in this case)

using randomized sampling. Randomized sampling first builds a low-dimensional matrix that preserves the range information of the original matrix. Next, an orthonormal basis is extracted from the low-dimensional matrix. The basis of the low-dimensional matrix also spans the range of the original matrix. The underlying assumption here is that the matrix has a low-dimensional column space. This assumption fits our setting because degradation signals typically have a high degree of temporal correlation. Step 2 computes the SVD of the original matrix using the basis extracted from step 1.

The RLA algorithm assumes that the rank (the number of principal components) of the original matrix is known a priori. Suppose the rank of $\boldsymbol{S}$ is K, then any K linearly independent vectors drawn from its column space can span its range. Step 1 draws K independent vectors from the range of matrix $\boldsymbol{S}$. This is achieved by randomized sampling, which involves generating K Gaussian random vectors, $\{\boldsymbol{w}_i\}_{i=1}^{K}$. It can be shown that for any vector $\boldsymbol{w}_i$, the multiplication $\boldsymbol{S}\boldsymbol{w}_i$ lies in the range of $\boldsymbol{S}$. Since vectors $\{\boldsymbol{w}_i\}_{i=1}^{K}$ are randomly generated, they are almost surely in a general position such that no linear combination of these vectors falls into the null space of $\boldsymbol{S}$. Consequently, the sample vectors $\{\boldsymbol{S}\boldsymbol{w}_i\}_{i=1}^{K}$ are linearly independent. Thus, $\{\boldsymbol{S}\boldsymbol{w}_i\}_{i=1}^{K}$ spans the range of $\boldsymbol{S}$. Note $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K)$ and $\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{W} = (\boldsymbol{S}\boldsymbol{w}_1, \boldsymbol{S}\boldsymbol{w}_2, \ldots, \boldsymbol{S}\boldsymbol{w}_K)$, then matrix $\boldsymbol{Y}$ has the same range as the original matrix $\boldsymbol{S}$. An orthonormal basis (denoted by $\boldsymbol{Q}$ in Table 3.1) for $\boldsymbol{S}$ can be obtained by applying SVD or QR decomposition on $\boldsymbol{Y}$.

If the rank of matrix $\boldsymbol{S}$ is exactly K, then the basis extracted in Step 1 exactly spans the range of $\boldsymbol{S}$. However, in reality, the rank of $\boldsymbol{S}$, denoted by R, is usually larger than K. Recall that in FPCA, we use the first K principal components to approximate matrix $\boldsymbol{S}$. The remaining $R-K$ principal components are often dropped because they do not capture much information. However, these $R-K$ principal components tend to shift the sample vectors, $\{\boldsymbol{S}\boldsymbol{w}_i\}_{i=1}^{K}$, outside the range spanned by the first K principal components of matrix $\boldsymbol{S}$, and hence may affect the accuracy of the RLA algorithm. This issue is addressed in two ways: oversampling and power interation. Oversampling reduces the probability of the

sample vectors being affected by the remaining $R-K$ principal components. It works by generating an additional r random sample vectors from the range of $\boldsymbol{S}$. That is, a Gaussian matrix $\boldsymbol{W}$ with $K+r$ instead of K columns is generated in ① of Step 1, where r is called the oversampling parameter. Power interation increases the weight of the first K principal components when constructing the sample vectors. It works by alternately multiplying the sample vectors, $\boldsymbol{SW}$, with $\boldsymbol{S}$ and $\boldsymbol{S}^{\top}$. That is, $\boldsymbol{Y}=(\boldsymbol{SS}^{\top})^{q}\boldsymbol{SW}$, where q is the power iteration parameter. Usually, $q \leq 2, r \leq 10$ is sufficient for most applications [20].

After extracting a basis $\boldsymbol{Q}$ for the range of $\boldsymbol{S}$ in Step 1, Step 2 computes the SVD of matrix $\boldsymbol{S}$ by exploiting the basis $\boldsymbol{Q}$. The following proposition shows that substeps ④, ⑤, and ⑥ of the RLA algorithm approximate the SVD decomposition of the signal matrix $\boldsymbol{S}$.

Proposition 1 *Given matrix $\boldsymbol{S}$ and one group of its approximate orthonormal basis $\boldsymbol{Q}$, that is, $\boldsymbol{S} \approx \boldsymbol{QQ}^{\top}\boldsymbol{S}$, substeps ④, ⑤, and ⑥ in Table 3.1 compute the approximate SVD decomposition of matrix $\boldsymbol{S}$.*

Proof 1 $\boldsymbol{S} \approx \boldsymbol{QQ}^{\top}\boldsymbol{S}=\boldsymbol{QB}=\boldsymbol{Q}\tilde{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^{\top}=\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^{\top}$, *where* $\hat{\boldsymbol{U}}=\boldsymbol{Q}\tilde{\boldsymbol{U}}$.

Table 3.1: Randomized low-rank approximation algorithm.

Given the degradation signal matrix $\boldsymbol{S}_{N\times L}$, the principal component number K, an exponent q and a over sampling number $r, r \geq 2$, $(K+r) < \min\{N, L\}$, RLA computes an approximate factorization $\boldsymbol{S} \approx \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^{\top}$.
Step 1: ① Generate an $L \times (K+r)$ Gaussian matrix $\boldsymbol{W}$. ② Form $\boldsymbol{Y}=(\boldsymbol{SS}^{\top})^{q}\boldsymbol{SW}$ by multiplying alternately with $\boldsymbol{S}$ and $\boldsymbol{S}^{\top}$. ③ Construct a matrix $\boldsymbol{Q}$ whose columns form an orthonormal basis for the range of $\boldsymbol{Y}$ via SVD. **Step 2:** ④ Form $\boldsymbol{B}=\boldsymbol{Q}^{\top}\mathrm{S}$. ⑤ Compute an SVD of the small matrix: $\boldsymbol{B}=\tilde{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^{\top}$. ⑥ Set $\hat{\boldsymbol{U}}=\boldsymbol{Q}\tilde{\boldsymbol{U}}$.

In the RLA algorithm, the ordinary SVD operation is first applied to matrix $\boldsymbol{Y}$ which has a dimensionality of $N \times (K+r)$ in substep ②. Then it is applied to matrix $\boldsymbol{B}$ which

has a dimensionality of $(K + r) \times L$ in substep ⑤. Since $K + r \ll \min\{N, L\}$, the dimensionality of both matrix $\boldsymbol{Y}$ and $\boldsymbol{B}$ are much smaller than the dimensionality of $\boldsymbol{S}$, which is $N \times L$. As a result, the computational speed of RLA is much faster than applying SVD directly on matrix $\boldsymbol{S}$.

RLA approximates matrix $\boldsymbol{S}$ by only keeping its first K principal components using randomized sampling. As mentioned earlier, the rank of matrix $\boldsymbol{S}$ is denoted by R, which is usually larger than K. The approximation error comes from dropping the remaining $R - K$ principal components as well as randomized sampling technique itself. [20] gives a probabilistic error bound for RLA algorithm. If we denote the approximated matrix by $\hat{\boldsymbol{S}}_{(K)}$, that is $\hat{\boldsymbol{S}}_{(K)} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^{\top}$, the probabilistic error bound for RLA is expressed by

$$\mathbb{E}[\|\boldsymbol{S} - \hat{\boldsymbol{S}}_{(K)}\|] \leq \left[\left(1 + \sqrt{\frac{K}{r-1}}\right)\sigma_{K+1}^{2q+1}(\boldsymbol{S}) + \frac{e\sqrt{K+r}}{r}\left(\sum_{j>K}\sigma_j^{2(2q+1)}(\boldsymbol{S})\right)^{\frac{1}{2}}\right]^{\frac{1}{2q+1}} \tag{3.17}$$

where $\|\cdot\|$ is the spectral norm.

3.5.2 Selecting the number of principal components of FPCA

In practice, finding the number of significant principal components K used in SVD is important, yet challenging. Typically, K is determined by calculating the fraction of the variance explained (FVE) by the selected singular vectors, and it is determined after computing the entire set of singular values and vectors. Let F_p define the FVE by the first p principal components, it is calculated by:

$$F_p = \frac{\sum_{j=1}^{p}\sigma_j^2(\boldsymbol{S})}{\sum_{j=1}^{\min\{N,L\}}\sigma_j^2(\boldsymbol{S})}, p = 1, \ldots, \min\{N, L\}. \tag{3.18}$$

Given an FVE threshold D, say 0.9, the number of principal components is determined by $K = \inf_p\{F_p \geq D\}$. In the RLA setting, however, the challenge is that F_p cannot

be computed, since not the whole set of $\sigma_j^2(\boldsymbol{S})$ are known. To address this challenge, we propose the following strategy. We first run the RLA algorithm by setting the rank as a small initial guess value G, and estimating the singular values, $\sigma_1(\hat{\boldsymbol{S}}_{(G)}), \ldots, \sigma_G(\hat{\boldsymbol{S}}_{(G)})$. Next, we calculate FVE as follows:

$$F_p = \frac{\sum_{j=1}^{p} \sigma_j^2(\hat{\boldsymbol{S}}_{(G)})}{\|\boldsymbol{S}\|_F^2}, p = 1, \ldots, G, \tag{3.19}$$

where $\sigma_j^2(\hat{\boldsymbol{S}}_{(G)})$ is the jth singular value of the approximated matrix in RLA, and $\|\boldsymbol{S}\|_F^2$ is the Frobenius norm of the signal matrix. Note that the numerator is the variation explained by the first p singular vectors and the denominator represents the total variation of the signal matrix. The number of principal components is determined by $K = \inf_p\{F_p \geq D\}$. If none of the $F_p, p = 1, \ldots, G$ values satisfy $\{F_p \geq D\}$, then G is increased, and the whole procedure is repeated until the number of principal components K is found. Usually, the guess rank G can be set as $G = 0.2 * \min\{N, L\}$ and added by $0.1 * \min\{N, L\}$ each time until the right K is found.

3.6 Simulation study

This section presents a simulation study to test our modeling framework. Two metrics are used to evaluate the performance of our approaches: computational time and accuracy of estimating the RUL. Specifically, we test the performance of the *Multivariate* and the *Hierarchical* FPCA methodologies. We also evaluate the performance of the same two techniques after incorporating the RLA algorithm, which we designate as "Multivariate FPCA+RLA" and "Hierarchical FPCA+RLA". We also study the impact of the sensors on computational time. We consider three settings: 100, 500 and $1,000$ sensors.

The performance of the four approaches mentioned above is compared to two baseline methods. We refer to the first baseline method as "Individual FPCA (best)". This approach models each sensor signal independently using FPCA. The resulting FPC-scores of each

sensor are used to construct separate (log)-location-scale regression models, each of which is then used to predict an RUL. The RUL with the least prediction error is reported. The second benchmark, which we designate as "Individual FPCA (mean)", is similar to "Individual (best)" except that the predicted RUL of the unit is obtained by averaging over the individual predictions of each sensor. Prediction errors are evaluated at the following life percentiles: 30%, 60%, and 90%. Note that the prediction errors are evaluated retroactively after failure has occurred to see how well each method performed. The prediction errors for each unit are computed using equation (2.25). The simulation scenarios are performed using MATLAB 2012b in a 64-bit Unix system with the Xeon X5560 CPU @2.80 GHz processor and 148.0 GB RAM.

3.6.1 Generating simulated degradation signals

In this simulation study, we consider 200 identical units, each of which is assumed to be monitored by $1,000$ sensors. The degradation signals are simulated by first simulating the underlying degradation path for all the 200 units using the following model: $s_i(t) = \theta_i / \ln(t)$, for $i = 1, \ldots, 200$, where $\theta_i \sim N(1, 0.3^2)$ and $0 \leqslant t < 1$. The TTF is computed as the first time point the underlying degradation trajectory, $s_i(t)$, crosses the failure threshold D, where $D = 5$. The histogram of TTFs is shown in Figure 3.2. Next, we generate degradation signals corresponding to each sensor by $s_{i,p}(t) = \theta_{i,p} / \ln(t) + \epsilon_{i,p}(t)$, where $p = 1, \ldots, 1,000$ and $\epsilon_{i,p}(t) \sim N(0, 1)$. To mimic practical applications, in which the degradation signal of each sensor may have a unique correlation with the underlying physical degradation, we generate $\theta_{i,p}$ from the following conditional distribution $\theta_{i,p}|\theta_i \sim N(1, 0.3^2)$ such that the correlation between $\theta_{i,p}$ and θ_i is a random variable chosen from a uniform distribution $\mathcal{U}[0.5, 0.95]$. It can be easily shown that the relationship between the TTFs and $s_i(t)$ can be captured by a (log)-location-scale regression model with lognormal distribution. The simulation procedure is replicated 100 times. For each replication, 100 units are randomly chosen for training and the remaining 100 units are used for testing. For

the FPCA framework, the number of principal components are chosen by setting FVE to 0.99.

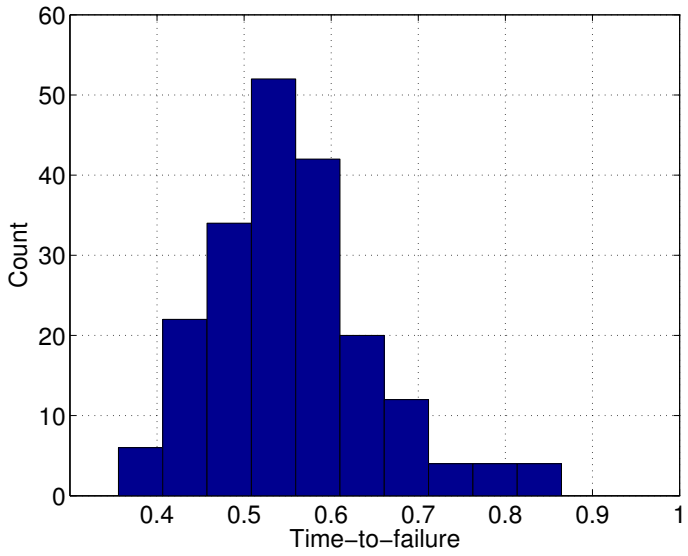


Figure 3.2: The histogram of TTFs of the simulated dataset.

3.6.2 Results and analysis

The average computation time for predicting the RULs of 100 units is reported in Table 3.2. Results indicate that the compute time of "Multivariate FPCA + RLA" is significantly less than that of "Multivariate FPCA". We observe that this difference becomes more profound as the number of sensors increases, which is attributed to incorporating the RLA algorithm since all other parameters remained the same. The gain in computational efficiency for "Hierarchical FPCA" is not as much as what was observed with the "Multivariate FPCA". This is because "Hierarchical FPCA" applies RLA on individual signals from each sensor, thus the degradation matrix is much smaller for the *Hierarchical* case relative to the *Multivariate* one. As a result, the compute times associated with using regular SVD versus RLA is not as significant as the "Multivariate FPCA".

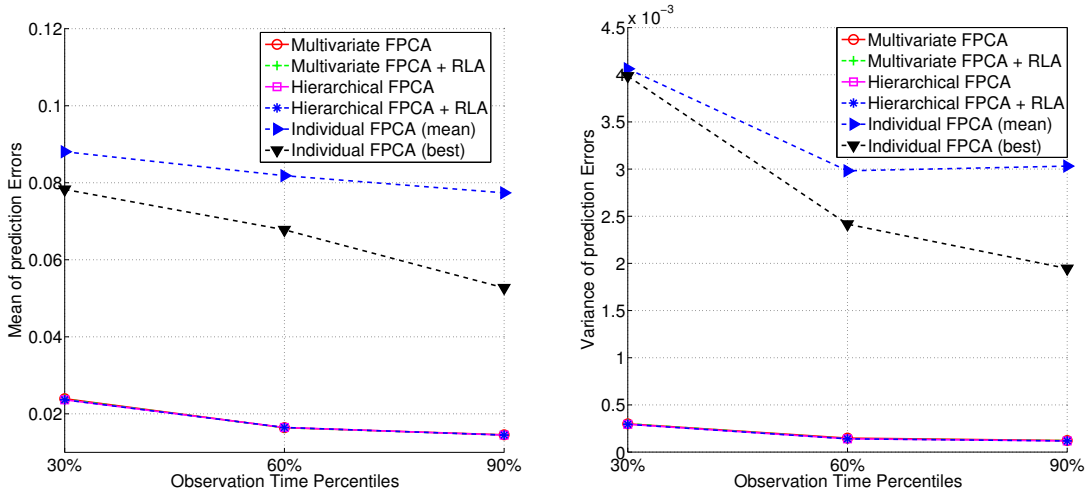
Figure 3.3 presents that the mean and variance of the prediction errors of the different modeling approaches and benchmarks. We observe that the "Multivariate FPCA +

Table 3.2: Average computation time for various models (unit: second).

	100 sensors	500 sensors	1,000 sensors
Multivariate FPCA	13	362	1,437
Multivariate FPCA + RLA	1	3	10
Hierarchical FPCA	3	11	22
Hierarchical FPCA + RLA	1	2	5

RLA" and "Multivariate FPCA" are similar. This observation implies that the approximated decomposition using the RLA algorithm does not compromise the prediction accuracy. Therefore, RLA can speed up the prognostics process without affecting its predictive accuracy. A similar phenomenon can also be observed in the "Hierarchical FPCA + RLA" and "Hierarchical FPCA". By comparing "Multivariate FPCA + RLA" and "Hierarchical FPCA + RLA" with "Individual FPCA (mean)" and "Individual FPCA (best)", it can be seen that the fusion-based methods that capture the interdependencies among the various degradation signals outperform other methods tend to focus on individually modeling each degradation signal, and either aggregating the resulting RULs from each model or choosing the best prediction. For example, while the mean prediction errors of the RUL at 60% for fusion-based methods are around 1.8%, those of the "Individual FPCA" methods are at 7% and above. The plots in Figure 3.3 also reveal that prediction accuracy improves with the life percentile. This can be explained by the fact that new observations tend to improve prediction accuracy, in general.

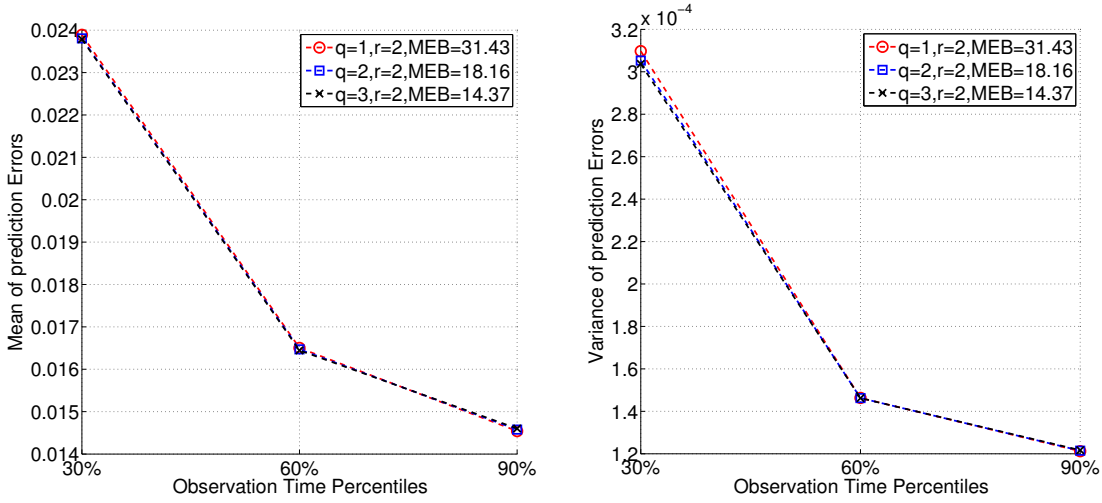
The error bound of the RLA has a significant impact on the performance of our methodology, specifically the prediction accuracy. As discussed in Section 3.5, the error bound of the RLA algorithm is affected by the oversampling parameter r and the power exponent parameter q. In this section, we conduct a sensitivity analysis on the choices of r and q. We design two simulation scenarios where we fix $r = 2$ and vary $q = 1, 2$ and 3; and another where we fix $q = 2$, and vary $r = 0.1K$, $0.2K$ and $0.3K$, where K is the number of principal components. In both scenarios, we choose "Multivariate FPCA + RLA" for evaluating performance. The mean and variance of prediction errors along with the mean error bounds



(a) mean (b) variance

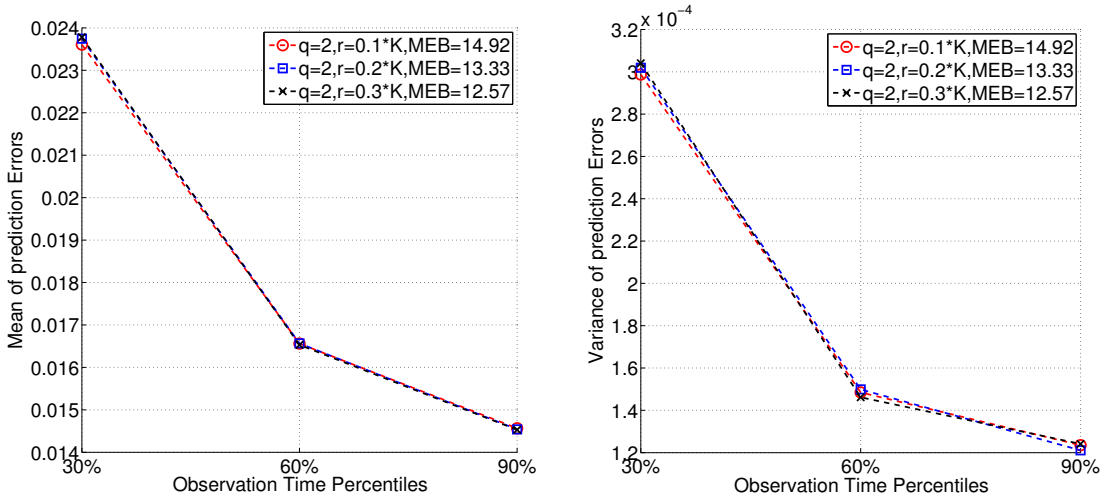
Figure 3.3: Mean and variance of prediction errors for the lognormal model.

(MEB) of each parameter combination are plotted in Figures 3.4 and 3.5. The figures show that although the change in r and q parameters affects the error bound, prediction errors remain relatively unchanged.



(a) mean (b) variance

Figure 3.4: Mean and variance of prediction errors for fixed r and varying q.



(a) mean (b) variance

Figure 3.5: Mean and variance of prediction errors for fixed q and varying r.

3.7 Case study

In this section, a multi-sensor degradation dataset from an aircraft turbofan engine simulated by NASA is used to evaluate the effectiveness of our three-step multi-sensor prognostic model. The data set, available from [11], comprises the following; (1) degradation signals from 100 *training* engines that were run to failure, (2) degradation signals from an additional 100 *test* engines whose operation was prematurely terminated at random time points prior to their failure time, and (3) the real TTFs of the 100 *training* engines and 100 *test* engines, all of which were monitored by 21 sensors. More detailed introduction on the dataset can be found in [11].

In this case study, we compare the performance of our methods "Multivariate FPCA + RLA" and "Hierarchical FPCA + RLA" with four benchmarks. In addition to "Individual FPCA (mean)" and "Individual FPCA (best)", we also include the health index methodology proposed by the authors of [32], who also used the degradation dataset. They visually removed ten sensors with flat signals that contained no information about engine degradation and created a composite health index by fusing the remaining 11 individual sensor

signals. Then they used the computed health indexes as a single synthesized degradation signal to predict the TTF of the engines. We designate this methodology "Health index (11 sensors)". Furthermore, as another benchmark, we also apply the health index model of [32] to all 21 sensors, denoted "Health index." In this case study, the number of principal components are chosen by setting FVE to 0.99.

3.7.1 Model selection and validation

We begin by selecting an appropriate (log)-location-scale regression model using the AIC criterion shown in the equation below:

$$AIC = -2\log(L) + 2p, \tag{3.20}$$

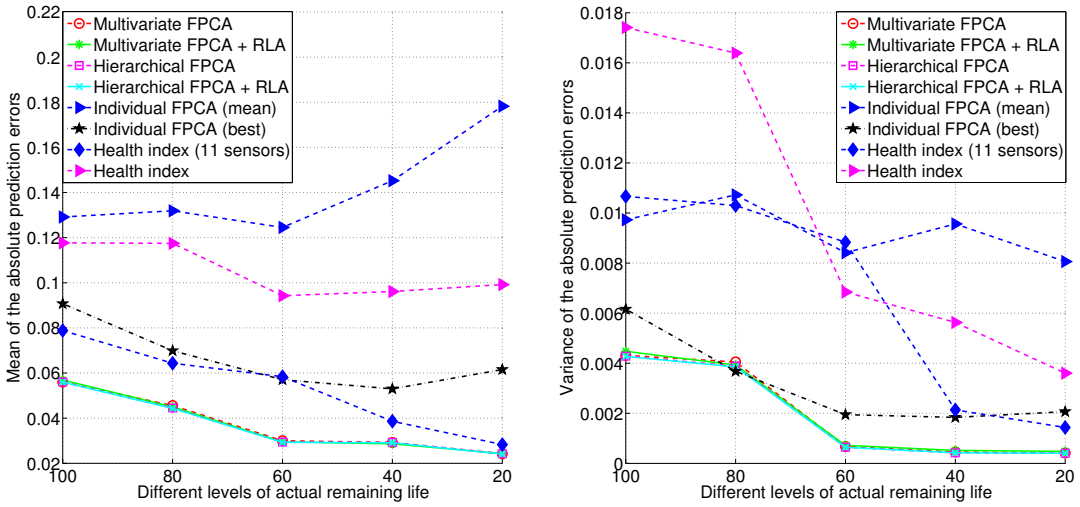
where $\log(L)$ is the maximum value of the log-likelihood function and p is the number of parameters in the model. Different (log)-location-scale regression models are applied to the training dataset, and the model with the lowest AIC value will be selected. To be specific, we first truncate all the degradation signals from the training dataset by only keeping their observations on time domain $[0, 128]$ epochs. Since the shortest TTF for the 100 engines in the training dataset is 128 epochs, the truncation can guarantee that all the training degradation signals share the same time domain, and hence our signal fusion methods can work (see Section 3.4 for details). Next, *Multivariate* FPCA is applied to the truncated degradation signals to extract fused features. These features are then regressed against TTFs by using different (log)-location-scale regression models. Four (log)-location-scale regression models commonly used in reliability engineering are tested: *normal regression*, *lognormal regression*, *smallest extreme value regression* and *Weibull regression*. The calculated AIC values for the four models are 993.8650, 947.8420, 1021.9283 and 968.3787 respectively, which suggests the lognormal regression model.

Next, we apply a goodness-of-fit test to check how the *lognormal regression* model fits the data. Specifically, we use both *Multivariate* FPCA and *Hierarchical* FPCA to extract

features from the truncated training degradation signals. Then, the features are regressed against the logarithmic TTF using a normal regression model. The adjusted R-squared for the model using features from *Multivariate* FPCA and *Hierarchical* FPCA is 93.59% and 93.12% respectively, which indicates that *lognormal regression* model fits the dataset well.

3.7.2 RUL prediction

In this subsection, we analyze the degradation dataset with our methods as well as the benchmarks and record the estimated RULs of the 100 test engines. Figure 3.6 reports the prediction errors against various levels of actual remaining lifetime the same way as they were reported in [32]. As illustrated in the figure, the reported prediction errors corresponding to the remaining lifetimes 100 represent the average prediction errors of the engines whose remaining lifetimes are equal to or shorter than 100 epochs.



(a) mean (b) variance

Figure 3.6: Mean and variance of prediction errors for the aircraft engine dataset.

Figure 3.6 reveals no significant difference between "Multivariate FPCA + RLA" and "Multivariate FPCA" or between "Hierarchical FPCA + RLA" and "Hierarchical FPCA", suggesting that the RLA algorithm does not compromise prediction performance. Further-

more, as expected, since the fusion-based methods capture the inter-relationship of signals, they demonstrate more accurate and precise predictions than "Individual FPCA" methods.

As the "Health index (11 sensors)" removes sensors with flat and corrupted signals prior to analysis, it outperforms "Health index" that focuses on all 21 sensor signals in terms of both the mean and the variance of prediction errors. "Health index (11 sensors)", however, cannot perform as well as our fusion-based methods. For example, at level 60, the mean prediction errors of our fusion-based methods are 0.03, whereas the mean prediction error for "Health index (11 sensors)" is 0.06. This performance gap decreases as the remaining life increases. In addition, the variance of prediction errors of "Health index (11 sensors)" is much larger overall than that of fusion-based methods, implying that fusion-based methods generated more precise predictions. Finally, we compute the computation times for "Multivariate FPCA + RLA", "Multivariate FPCA", "Hierarchical FPCA + RLA" and "Hierarchical FPCA": 58.2s, 291.5s, 200.9s and 204.1s, respectively. The reported computation times again indicate that the RLA can significantly expedite the proposed prognostics methodology. In short, based on the reported results, our methods outperform other benchmarks in terms of prediction accuracy and precision, as well as computation time.

3.8 Conclusion

Complex engineering systems are often monitored by a large number of sensors that generate massive amounts of degradation data. Building a prognostic model by utilizing such large-scale datasets poses two significant analytical challenges: how to effectively fuse the degradation signals from numerous sensors and how to make the model scalable to the large data size. To address the two challenges, this chapter presented a scalable semi-parametric statistical framework that utilizes multistream signals for predicting (in near real-time) the RULs of partially degraded systems. Our method first develops two multi-stream signal fusion algorithms, *Multivariate* FPCA and *Hierarchical* FPCA, to effectively

fuse the degradation signals from various sensors. Both of the fusion algorithms are capable of capturing the cross-correlation among different sensors, reducing the dimensionality and providing fused features. Next, with the fused features, we built an adaptive functional (log)-location-scale regression model for the dynamic prediction of RULs. In order to address the computational challenge, our method incorporated a RLA algorithm, which can help to speed up matrix decomposition for *Multivariate* FPCA and *Hierarchical* FPCA but without affecting the prediction accuracy.

Referring to a simulated dataset and a aircraft turbofan engine dataset from the NASA repository, we tested the performance of our method and drew several conclusions. One is that the RLA algorithm in our prognostic models can dramatically speed up the prognostics process without sacrificing its prediction accuracy. For example, while Table 3.2 showed that the computation time of "Multivariate FPCA + RLA" is much shorter than that of "Multivariate FPCA", Figure 3.3 showed that the mean and the variance of the prediction errors for "Multivariate FPCA + RLA" and "Multivariate FPCA" are very close, indicating that the approximated decomposition by the RLA algorithm does not significantly affect prediction accuracy. We draw same conclusion by comparing the mean and the variance of prediction errors as well as the computation time for "Hierarchical FPCA + RLA" and "Hierarchical FPCA".

We also concluded that our prognostic models are not sensitive to the approximation parameters of the RLA algorithm. The approximation error bound of the RLA is controlled by two parameters: an exponent q and an oversampling number r. Two scenarios of sensitivity analysis, that is, fixed q and varying r, fixed r and varying q, were implemented. The mean and the variance of prediction errors for both scenarios (Figures 3.4 and 3.5) showed that although the change in the r and q parameters affects the error bound, prediction errors are robust to these changes.

Another conclusion is that our fusion-based methods outperform other benchmarks in terms of both prediction accuracy and precision. First, our models outperformed other

benchmarks that individually model degradation signals. Figure 3.3 in the simulation and Figure 3.6 in the case study showed that both the mean and the variance of prediction errors for our "Multivariate FPCA + RLA" and "Hierarchical FPCA + RLA" models are significantly smaller than those of "Individual FPCA (mean)" and "Individual FPCA (best)". We believe this results from our fusion-based models taking signal cross-correlation into consideration. Furthermore, we compared our methods with the health index methodology proposed by [32], which used the same degradation dataset. The results showed that both our "Multivariate FPCA + RLA" and "Hierarchical FPCA + RLA" models outperform the health index method in terms of the mean and the variance of prediction errors.

CHAPTER 4
AN ADAPTIVE FUNCTIONAL REGRESSION-BASED PROGNOSTIC MODEL FOR APPLICATIONS WITH MISSING DATA

4.1 Introduction

Prognostic degradation models focus on characterizing how degradation signals (condition-based sensor signals) evolve over time. Typically, the goal is to estimate lifetime, or in our case, predict and update remaining lifetime in real-time. A large number of degradation models have been proposed in the literature, such as the ones presented in [70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 52, 81, 82, 83, 1, 35, 84]. The key objective of these models is to predict failure or explicitly estimate remaining useful lifetimes. However, the effectiveness of these models relies primarily on the fidelity of estimating their parameters. Parameter estimation is often driven by the characteristics and the quality of historical data (also known as, training data). For example, most models assume that historical degradation signals are completely and, for all practical purposes, continuously observed from an "as good as new" state up to the point of failure. In reality, however, continuous or frequent observations of degradation is not always possible nor economical. Examples of such scenarios include monitoring cracks on gas turbine blades that require shutting down the turbine or assessing the concentration of dissolved gases in transformers. Other examples may involve sensor failure or disconnection. Thus, in practice, it is more likely that degradation signals are observed randomly or at intermittent points in time resulting in sparse or fragmented signal observations as illustrated in Figure 1.4.

If parametric models are used to model signals with such high levels of missing data, it is likely that the available data will not be enough to accurately identify a suitable trend or general path for the degradation signals. In this chapter, we utilize a semi-parametric ap-

proach to develop a prognostic degradation model for sparse and fragmented signals. First, Functional Principal Component Analysis (FPCA) is used to identify key features of the incomplete signals. FPCA is a nonparametric Functional Data Analysis (FDA) technique that identifies the important sources of pattern variation among functional data. One important benefit of FPCA is that it provides a low-dimensional and parsimonious representation of each curve by reducing it to a set of functional principal components scores (i.e., FPC-scores), which were referred to earlier as signal features. The FPC-scores estimated using signals with missing observations are likely to be similar to those that would have been estimated if all observation were present. Of course, as the level of missing data increases so does the difference between these scores. Once the signal features are extracted, an adaptive functional regression model is used to model the relationship between the FPC-scores and historical Times-to-Failure (TTFs). Functional regression is an extension of ordinary regression that accounts for the case where predictors are random functions and responses are scalars or functions [17]. A popular approach for implementing functional regression with scalar responses is to represent predictor functions with FPC-scores, and fit an ordinary regression model on these scores and the response scalars (see for example [17, 85, 47, 86, 48]).

Our proposed framework also provides a means to incorporate in-situ signals observed from partially degraded components in the field in order to to update the model. The updated model is then used to revise the predicted remaining lifetimes.

The remainder of the chapter is organized as follows: Section 4.2 reviews some of the relevant nonparametric modeling approaches. Section 4.3 discusses the development of the degradation model. In Section 4.4, we introduce the functional regression model that is used for predicting residual lifetime. An adaptive approach for updating the residual life predictions is discussed in Section 4.5. We then evaluate the performance of our methodology using real-world crack growth degradation signals in Section 4.6, and real-world vibration-based bearing degradation signals in Section 4.7. Finally, conclusions and future

research are presented in Section 4.8.

4.2 Literature review

Many degradation modeling approaches have been developed in the literature. Some examples involve the use of neural networks and fuzzy logic models [73, 78, 80, 52], Kalman and particle filtering techniques [81, 82, 83], statistical degradation models [77, 79, 1, 35], and various types of stochastic processes, such as the Wiener and Gamma processes [74, 75, 76, 84]. However, most of the existing models focus only on complete degradatoin signals. Some research efforts have focused on modeling sparse and irregularly observed degradation data (also known as *longitudinal data* in the bioinformatics and medical literature) using FDA [87, 88, 89, 49, 90, 91, 92, 63, 64, 93]. For example, in [49] the authors used FPCA for applications with sparse longitudinal data. They assumed that repeated measurements exhibit an underlying smooth random (subject-specific) trajectory plus measurement errors. They proposed the PACE approach (principal components analysis through conditional expectation) to extend the applicability of FPCA to situations with sparse longitudinal data. In [88], the authors introduced a latent Gaussian process model for sparse longitudinal data measurements observed at irregular intervals. They also used FPCA and illustrated their model on biliary cirrhosis data. [91] applied FPCA to model sparse and noise-contaminated longitudinal data. This work was later extended in [90] to address sparse and unsynchronized longitudinal data. In [89], the authors incorporated FPC- scores in a reduced rank mixed effects framework. In [92], the authors proposed a support vector machine approach that used FPCA to perform multi-category classification of sparse data with a multi-category response.

The underlying theme of most of the research mentioned above is that different realizations of the the longitudinal data (which are analogous to our signals) are assumed to share the same time domain. In other words, the start and end points are the same for all the realizations. As pointed out in [63, 64, 93], one of the major limitations of using FPCA

is that it indeed requires that each observed curve shares the same time domain. Unfortunately, this is not the case for most engineering systems as they are often shutdown, taken off-line for repair, or replaced once their degradation signals reach an alarm threshold (see for example ISO 2372 and 10816 for machine vibration). In other words, degradation signal are typically truncated at the threshold beyond which no subsequent observations can be acquired. In such scenarios, using FPCA results in a significantly biased estimate of the mean and covariance functions due to the fact unobserved data beyond the threshold [63]. This aspect is a unique difference between most of the related literature and the work proposed in this chapter.

There have been some recent attempts to tackle this problem. For example, [64] proposed a procedure that relied on axis transformation. (Instead of plotting the degradation level on the y-axis with time on the x-axis, the axes were reversed.) However, this approach created another problem especially with noisy signals whereby a the same degradation level may correspond to multiple time stamps. From a practical standpoint, this was infeasible. Thus, the approach was limited to strictly monotonic signals with very low noise levels. [93] developed a functional time warping approach that can synchronize the truncated degradation signals, but it focuses only on complete degradation signals. Unlike existing work, our modeling approach is significantly more general in that it can be applied to various types of signals, and has no restrictions on the signal-to-noise ratio.

4.3 Degradation model development

As mentioned earlier, we consider a problem setting wherein historical degradation signals contain a significant level of missing data. Specifically, we focus on two types of signals, sparsely observed and fragmented signals, as illustrated earlier in Figure 1.4. Furthermore, signals are only observed up to a predetermined failure/replacement threshold. We assume that degradation signals are independent realizations of a smooth random function over a bounded time domain $[0, M]$. M is the maximum observation time point, which is assumed

to be finite since any industrial application has a finite time-of-failure. (Hypothetically, this is the degradation signal of the component with the longest possible lifetime.) Taking observation error into account, our approach is to model the amplitude of the degradation signal of the ith component, $S_i(t)$, as follows;

$$S_i(t) = \mu(t) + X_i(t) + \epsilon_i(t), \tag{4.1}$$

where $i = 1, \ldots, n$, and n represents the number of components, $\mu(t)$ represents the underlying trend of the degradation signal that is common to the entire population and is assumed to be deterministic, $X_i(t)$ represent stochastic deviations from the underlying trend due to the inherent variability in the degradation rates of the components. $X_i(t)$ are assumed to follow a stochastic process with mean zero and covariance function $cov((X(t), X(t')) = C(t, t')$. Finally, $\epsilon_i(t)$ are independent and identically distributed observation errors with mean zero and variance σ^2. $X_i(t)$ and $\epsilon_i(t)$ are assumed to be independent.

Using the Mercer's theorem [94], the covariance function $C(t, t')$ can be expanded as shown below,

$$C(t, t') = \sum_{k=1}^{\infty} \lambda_k \phi_k(t) \phi_k(t'),\ t, t' \in [0, M], \tag{4.2}$$

where $\phi_k(t)$ for $k = 1, 2, \ldots$ are the eigenfunctions and $\lambda_1 \geq \lambda_2 \geq \ldots$, are the ordered nonnegative eigenvalues.

Since the eigenfunctions are orthogonal and form a functional basis, $X_i(t)$ can be expressed as follows;

$$X_i(t) = \sum_{k=1}^{\infty} \xi_{ik} \phi_k(t), \tag{4.3}$$

where ξ_{ik} for $k = 1, 2, \ldots$ are the FPC-scores. These scores are independent random variables with mean $E[\xi_{ik}] = 0$ and variance $E[\xi_{ik}^2] = \lambda_k$. Thus, the amplitude of the

degradation signal can now be rewritten as follows:

$$S_i(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(t) + \epsilon_i(t). \tag{4.4}$$

Generally, the eigenvalues λ_k for $k = 1, 2, \ldots$ decrease to zero rapidly. Therefore, it is reasonable to assume that an appropriate K can always be chosen to approximate $X_i(t)$. For example, Equation (4.4) can be truncated by selecting K to minimize the Akaike information criterion (i.e., AIC) defined by [48]. AIC is a widely used model selection criterion, which deals with the trade-off between the goodness-of-fit and the complexity of the model. In our case, we select the first k eigen functions to approximate Equation 4.4 and calculate their corresponding AIC values, i.e., $\text{AIC}_k(k = 1, 2, \ldots; \lambda_k > 0)$. Then the k corresponds to the smallest AIC value is chosen to truncate our model. The (truncated) model that will be the basis of our work is expressed below by Equation 4.5:

$$S_i(t) = \mu(t) + \sum_{k=1}^{K} \xi_{ik}\phi_k(t) + \epsilon_i(t). \tag{4.5}$$

The different parts in Equation (4.5), i.e., $\mu(t)$, ξ_{ik}, $\phi_k(t)$ and $\epsilon_i(t)$, can be estimated using the historical degradation signals. Details of the estimation can be found in the appendix of this chapter.

4.4 Functional regression analysis

We now discuss the functional regression model used to estimate component lifetime. First, we denote the lifetime of the ith component as Y_i. By setting the scalar Y_i as response and degradation signals $S_i(t)$ as the predictor, we can establish a classic linear functional regression model [17]:

$$Y_i = \int_{[0,M]} \alpha(t) S_i(t) dt, \tag{4.6}$$

where the regression parameter function $\alpha(\cdot)$ is assumed to be smooth and square integrable over the interval $[0, M]$. Since $\phi_k(t)$ form a complete orthonormal basis for $k = 1, 2, \ldots,$ $\alpha(t)$ can be expanded in terms of these basis as follows; $\alpha(t) = \sum_{k=1}^{\infty} \beta_k \phi_k(t)$. Since $S_i(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(t) + \epsilon_i(t)$, we can now express the our functional regression model as follows;

$$\begin{aligned} Y_i &= \int_{[0,M]} \alpha(t)\mu(t)dt + \int_{[0,M]} \left\{\sum_{k=1}^{\infty} \beta_k \phi_k(t)\right\} \left\{\sum_{k=1}^{\infty} \xi_{ik}\phi_k(t)\right\} dt + \int_{[0,M]} \alpha(t)\epsilon_i(t)dt \\ &= \beta_0 + \sum_{k=1}^{\infty} \beta_k \xi_{ik} + \varepsilon_i, \end{aligned} \tag{4.7}$$

where $\beta_0 = \int_{[0,M]} \alpha(t)\mu(t)dt$ is the intercept, $\beta_k = \int_{[0,M]} \alpha(t)\phi_k(t)dt$ are regression coefficients, and $\varepsilon_i = \int_{[0,M]} \alpha(t)\epsilon_i(t)dt$ is the error term. Recall that we truncate $S_i(t)$ using a finite sum of K terms to approximate the infinite sum in Equation (4.5). Using this truncated form, the regression model can be expressed as,

$$Y_i = \beta_0 + \sum_{k=1}^{K} \beta_k \xi_{ik} + \varepsilon_i. \tag{4.8}$$

The coefficients β_k for $k = 1, \ldots, K$ can be estimated using least square estimation. We assume that there exists a database of historical degradation signals that we refer to as training signals. We also assume that the lifetimes of the components corresponding to the training signals are known a priori. The training signals are first used to estimate the FPCA scores as discussed in the appendix. Next, the regression model is estimated using the FPC-scores and the corresponding lifetimes. Note that more complicated functional regression models, for example, functional quadratic regression model or functional polynomial regression model, can be also used to mode the relationship between the lifetime Y_i and degradation signals $S_i(t)$.

The main objective of this framework is to enable predicting the residual lifetime of components that are still operating in the field. To do this, we propose an updating scheme whereby in-situ observations from the fielded components are leveraged to update the model based on the latest degradation states of the fielded components, thus improving the accuracy of remaining life predictions.

4.5 Estimating and updating remaining lifetimes

Degradation signals observed from fielded components provide a wealth of knowledge about their current degradation. Consequently, an inherent component of our framework is to provide the means to incorporate in-situ signals observed from the field and utilize them to provide more accurate predictions of remaining lifetime. To illustrate this updating scheme, recall that there are two sets of degradation signals, training and validation. The training signals are used to estimate the FPC-scores, and build a regression model using the scores and the TTFs of their corresponding signals. On the other hand, the set of validation signals are assumed to represent signals observed from fielded components. Next, consider a validation signal observed up to a specific time point. The FPC-scores of the partial validation signal are estimated using the eigenvectors calculated from the training set. The FPC-scores are then input to the regression model and a time-to-failure for the validation signal is estimated. As more signals are observed, the FPC-scores are revised and new a TTF is estimated.

As mentioned earlier, to correctly use FPCA, all the functional curves associated with the degradation signals must share the same time domain. However, the training and validation signals used during the updating scheme will not possess this characteristic–probably never will for engineering applications where a failure threshold is used to plane for maintenance or repair). Due to the existence of a failure threshold, the support over which the signals are observed will vary from one component to another. To address this limitation, we borrow an adaptive updating approach that was first presented by [49]).

To illustrate this approach, we first consider a validation signal observed up to time t^* (see Figure 2.2). We then select only the training signals that have survived up to time t^*. This subset of training signals is used to compute FPC-scores (along with the eigenvalues and eigenvectors) and estimate the functional regression model. Next, we estimate the FPC-scores of the validation signal (represented by the thicker curve in Figure 2.2). By applying the validation FPC-scores to the revised functional regression model, the lifetime of the "validation component" can be estimated. The remaining lifetime can be easily computed by subtracting the current operating time t^*.

As more observations become available, different subsets of the training degradation signals will be selected at different values of t^*. However, the same estimation process will be repeated at every new value of t^*. Figure 2.2 shows an example of how training signals are selected at different observation times of the validation signal. The dotted lines represent the entire set of training signals. The continuous part marks the part of the data that is used for estimating the functional regression model. The signal marked with a thick line represents the portion of the validation signal observed at time t^*.

In the following section, we evaluate the performance of our methodology using two different sets of degradation data. The first case study deals with crack growth data from several tests specimens of aluminum alloy and the second uses relatively noisier vibration-based data from a rotating machinery.

4.6 Case study of crack growth data

In this study, crack growth data first presented by [95] is used to test the proposed model. The data set consists of crack propagation measurements pertaining to 68 identical center-cracked aluminum plates that were tested under identical experimental conditions. The data consists of the number of cycles for discrete levels of crack length from 9.0 mm to 49.8 mm. Each crack growth signal consisted of 164 data points.

We randomly chose 58 signals as our training signals with the remaining 10 signals

for validation, i.e., representing signals observed in real-time from fielded components. The performance of our model was tested under three scenarios, (1) signals with complete observations, (2) signals with sparse observations, and (3) fragmented signals. For the complete signals, we use the original crack data since observations were made at a relatively high frequency. In the sparse scenario, we randomly sample 12 observations from each training and validation signal. In this scenario, model estimation and validation is conducted using the sparse signals, i.e., the sparse training signals are used for model estimation and the validation are used to evaluate performance. A similar approach is also used in the fragmented scenario except that the fragmented signals are generated by randomly sampling 3 fragments with 4 observations per fragment from each original signal (training and validation). A sample of the data is shown earlier in Figure 1.4. The evaluation process is replicated 100 times, and the performance of our model is evaluated by computing the prediction error at predetermined life percentiles. Prediction errors are calculated at different life percentiles using error Equation (2.25).

4.6.1 Results and analysis

The prediction errors for the three signal scenarios are evaluated at the following life percentiles: 10% (10% of the component's lifetime has passed at the point the prediction was made), 30%, 50% ,70% and 90%. Figures 4.1, 4.2, and 4.3 depict box plots of the prediction errors for the complete, sparse, and fragmented scenarios, respectively. Note that remaining life predictions are updated based on the validation signals that have been observed at each life percentile.

Overall, the complete signal scenario illustrated Figure 4.1 has the lowest mean prediction error. It also has the smallest confidence intervals across all life percentiles when compared to the other two signal scenarios. The box plots also demonstrate the effect of the updating process which can be clearly seen through the progressive reduction in the mean prediction error and corresponding variance. When comparing the prediction errors across

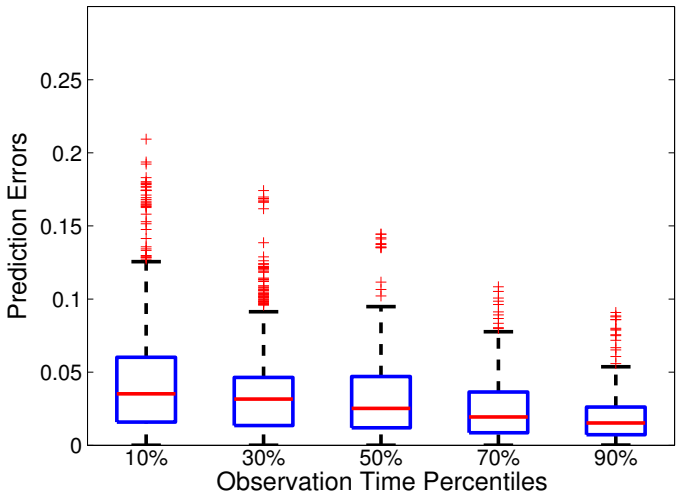


Figure 4.1: Prediction errors of the proposed model under the complete degradation signals scenario.

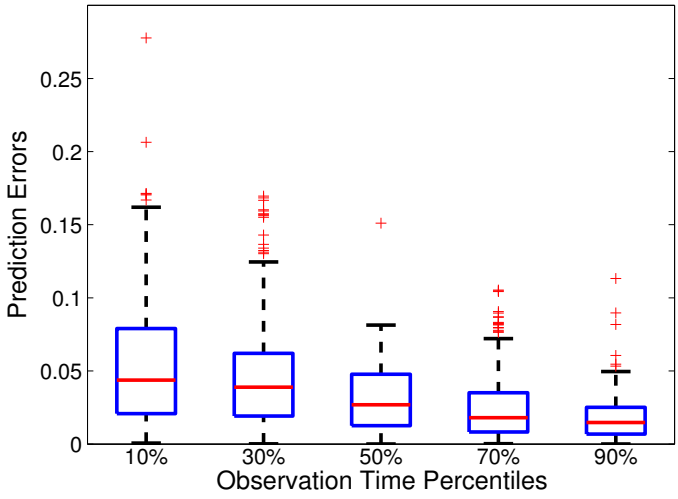


Figure 4.2: Prediction errors of the proposed model under the sparse degradation signals scenario.

the three signal scenarios, we observe that although significantly less signal observations were used in the sparse and fragmented scenarios, the proposed model still maintains relatively robust performance compared to complete scenario (compare Figure 4.1 with Figures 4.2 and 4.3). In fact, there seems to be no significant difference between the three signal scenarios at and beyond the $50th$ life percentile.

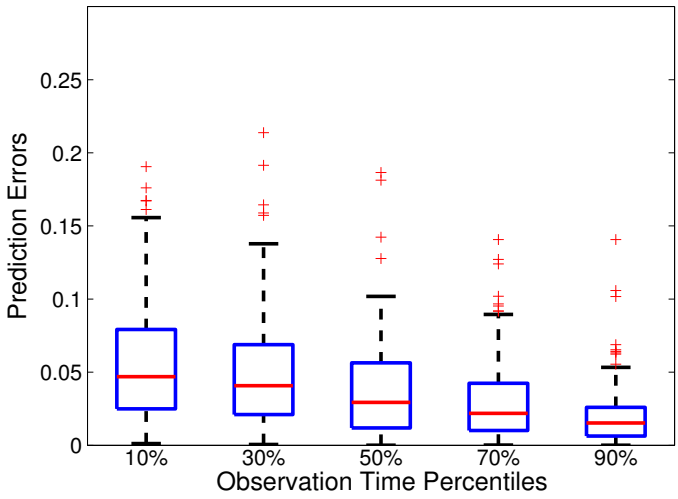


Figure 4.3: Prediction errors of the proposed model under the fragmented degradation signals scenario.

Based on these results, we believe that our modeling approach demonstrates significant potential for predicting remaining lifetimes of fielded, especially in applications with significant levels of missing data. This conclusion is of course governed by the characteristics of the signals. For example, results are especially promising in this example because the signals have a high signal-to-noise ratio. In the next case study, we evaluate the performance of our model using degradation signals with a significantly lower signal-to-noise ratio.

4.7 Case study of rotating machinery degradation

In this case study, the performance of our model is evaluated using vibration-based degradation signals generated from degradation of a rotating machinery. Specifically, the experimental test rig is designed to perform accelerated degradation tests on rolling element thrust bearings. Vibration signatures are used to monitor bearing degradation. A detailed description of the experimental setup, test conditions, and the degradation signals can be found in [96]. The degradation signals used in the study represent the average amplitude of the defective frequency and its first six harmonics over time. A bearing is considered to

have failed once the amplitude of its degradation signal crosses a pre-specified threshold of $0.025\ V_{rms}$ (which are mapped from industrial ISO standards). For the purpose of brevity, the reader is referred to [96] for additional details.

The data set consists of degradation signals for 31 bearings that were tested until failure. Observations were acquired every 2 minutes with lifetimes ranging between 12 to 36 hours. For the sparse scenario, a new training is created by randomly choosing 10 observations from each of the original complete signals. A similar process is also used to construct the degradation signals for the fragmented case. Three fragments are randomly chosen from each original signal, each fragment is consisting of 3 observations. Examples of sparse and fragmented bearing degradation signals are shown in plot "a" of Figures 4.4 and 4.5. A leave-one-out cross validation is performed by choosing one signal and using the remaining 30 for training and model estimation. Prediction errors are estimated using the same expression in Equation 2.25.

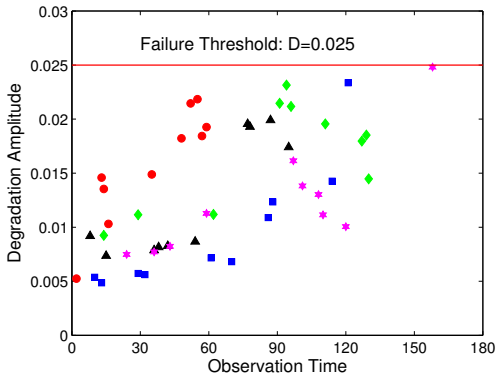
The above procedure starting with creating the database up to the cross validation and evaluating the prediction errors is repeated 20 times, resulting in 620 validation tests. This is performed for the complete, sparse, and fragmented signals. To benchmark the performance of our model, the sparse and fragmented signal scenarios are compared to two existing models (1) 'classical FPCA', and the (2) 'axis-transformation FPCA' model. With respect to the complete signal scenario, our model is benchmarked against the parametric exponential stochastic model [1] that was developed for this specific data set.

4.7.1 Results and analysis for degradation signals with missing data

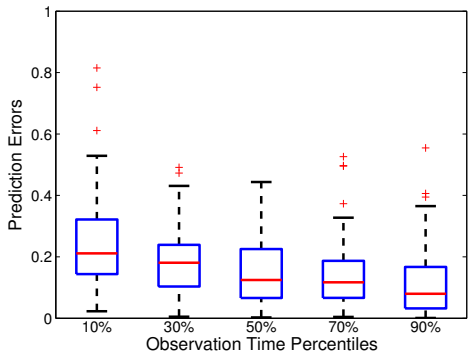
We now discuss the benchmark models used for the sparse and fragmented signals. The first benchmark model, which we define as 'classical FPCA', involves using FPCA in the classical sense. In other words, FPCA is used to fit a nonparametric model to data. The model is then used to estimate failure times and/or remaining lifetimes by extrapolation, given a predefined failure threshold. This is similar to the manner in which FPCA was

used in [87, 88, 89, 49]. The second benchmark is the 'axis-transformation FPCA' model proposed by [64], which was discussed in Section 4.2.

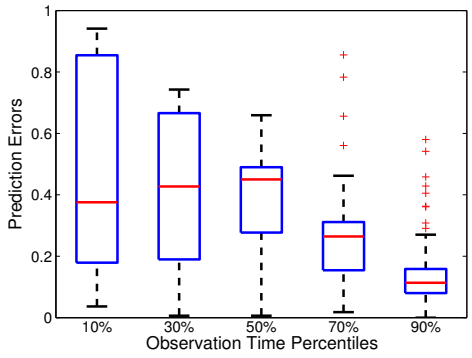
Prediction errors for the three models were evaluated using Equation 2.25 and are summarized in Figures 4.4 and 4.5. Results show that on average the prediction errors when using our functional regression model are relatively lower than the other two benchmark models. This is true for both the sparse and fragmented scenarios. As mentioned earlier, one of the main shortcomings of 'classical FPCA' is that it requires all signals to share the same time domain, a characteristic that is not necessarily satisfied especially that many equipment are shutdown for repair or maintenance once their degradation signals reach a pre-specified threshold.



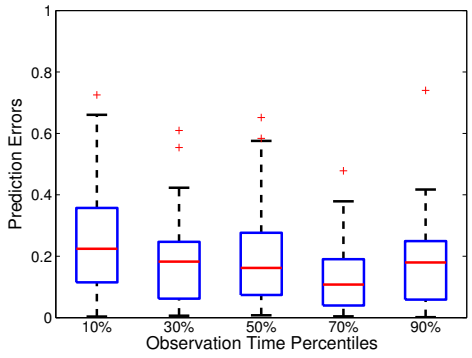
(a) A sample of sparse degradation signals



(b) Prediction errors using the functional regression model

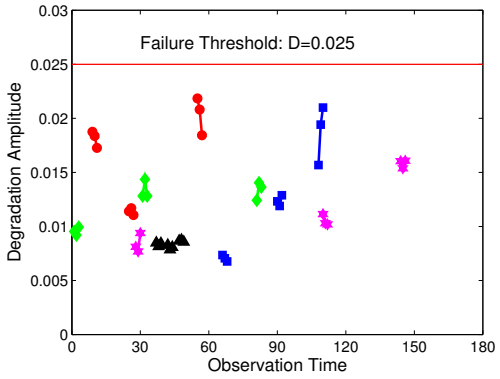


(c) Prediction error using a classical FPCA model

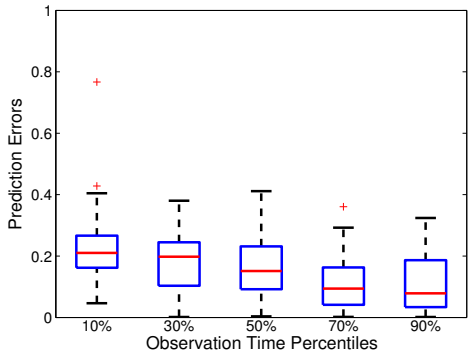


(d) Prediction errors using the axis-transformation FPCA model

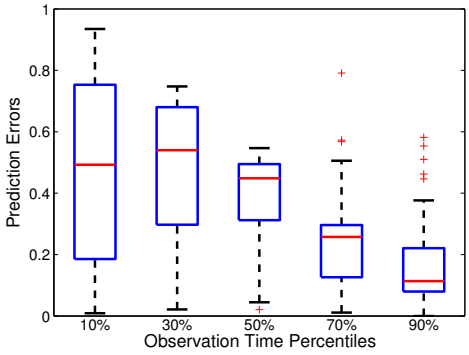
Figure 4.4: Plots of prediction errors under the sparse signals scenario.



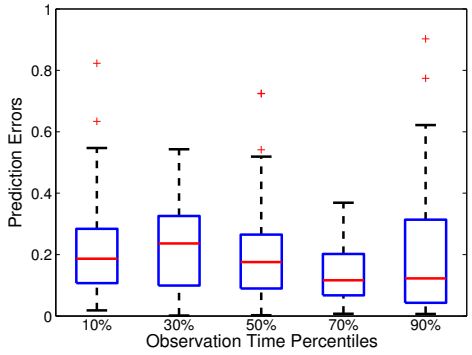
(a) example of fragmented degradation signals



(b) prediction error for functional regression



(c) prediction error for classical FPCA



(d) prediction error for the axis-transformation FPCA model

Figure 4.5: Plots of prediction errors under the fragmented signals scenario.

Prediction errors for the three models are evaluated in a similar manner as discussed in the previous case study using Equation 2.25. By observing the box plots of Figures 4.4 and 4.5, it can be seen that our approach performs better than 'classical FPCA'. Although FPCA has been used proven to work well in situations involving sparse data, one of its main shortcomings, as mentioned earlier, is the fact that it assumes all signals share the same time domain. From an practical viewpoint, this attribute does not hold in many engineering applications where components may be shutdown for repair/maintenance at different times once their degradation signals reach a pre-specified alarm or replacement threshold. By comparing the mean and variance of the prediction errors at the 10th, 30th, 50th, and 70th life percentiles, we see that both are significantly smaller in our model com-

pared to the 'classical FPCA' model. Even though the variance of the prediction errors at the 90th percentile in our approach is higher than 'classic-FPCA', the mean error remains relatively smaller. The increased variance phenomenon can be attributed to the fact that fewer components tend to have long lifetimes. This fact coupled with the sparsity of the data creates a significant level of variability at the 90th life percentile when using our approach.

By comparing Figures 4.4 and 4.5, we can see that our approach edges a little over the 'axis-transformation FPCA' model proposed by [64]. One observation that is clear is that the variance of the prediction errors for the benchmark model change from one life percentile to another. In addition they are consistently greater than those of our model. This observation is even more pronounced in the case of the fragmented signals (compare plots (b) and (d) of Figure 4.5). We believe this observation may be attributed to the fact that benchmark model is restricted to monotonic degradation signals. Consequently, for the relatively noisy signals used in this case study, the monotonicity condition is often violated resulting in poor performance of the model. Relative to the sparse signals, the effect of the noise is more visible in the fragmented case.

4.7.2 Results and analysis for complete signals

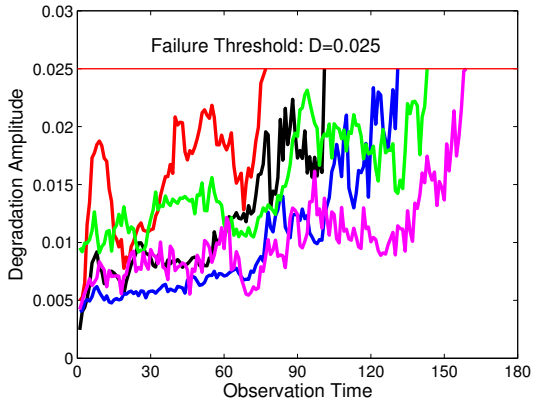
For the complete signal scenario, we compared our functional regression model to the exponential Brownian model (designated as 'exp-Brown') presented in [1]. The model consists of an random coefficients model with an exponential trend function and a Brownian motion error term. As mentioned earlier, this model was used as a benchmark because there was enough observations in the complete signals to identify a suitable degradation trend. In the same spirit, no parametric benchmark models are used for the sparse and fragmented scenarios since the significant levels of missing data are not enough to identify suitable trend functions. Box plots of the prediction errors are summarized in Figure 4.6. The plots show that the functional regression model performed well compared to the 'exp-Brown'

model. Variance was significantly smaller at the 10th, 30th, and 90th life percentiles when using the functional regression model. The benchmark model edges slightly over our model at the 50th life percentile in terms of the variance of the prediction error, but there seems to be no significant difference in terms of the mean of the prediction error. Based on the plots, it seems that our functional regression model still retains its effectiveness in the case of complete signals.

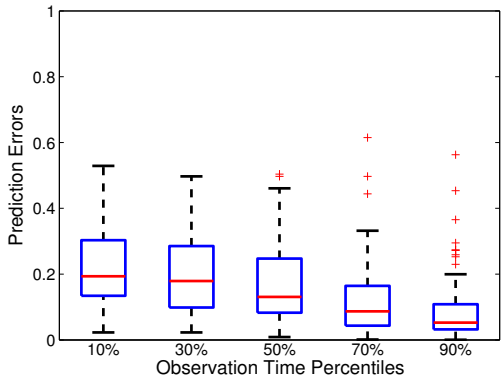
4.8 Summary

Many industrial applications provide the capability of real-time monitoring of performance and degradation. However, a common challenge in many industries is how to utilize this information given that often times there are missing data. In this chapter, we presented a functional regression model capable of predicting and updating, in real-time, the remaining lifetime of engineering systems. Our approach is best suited for applications in which degradation signals have different forms of missing data, i.e., sparse or fragmented data. This is especially beneficial since many parametric models no longer become a viable option due to the data sparsity.

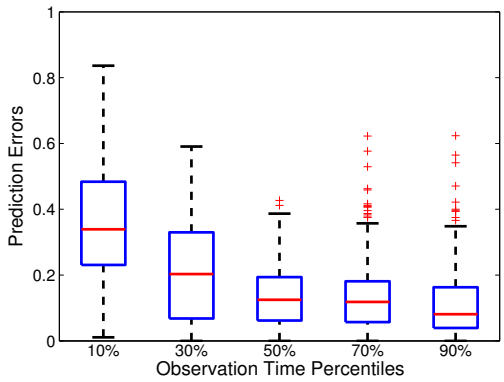
Our methodology was based on using FPCA to identify a general nonparametric trend for degradation signals pertaining to a population of similar components. An adaptive functional regression model was then used to model the relationship between the FPC-scores and the time-to-failure of the components. Real-time signals observed from validation components (assumed to be operating in the field) were incorporated into the model and used to update the predicted time-to-failure of each fielded component based on their unique degradation characteristics. The model was validated using two sets of degradation data, crack growth and bearing vibration data. The performance of the model was benchmarked against other nonparametric and parametric models. The investigation was performed for complete, sparse, and fragmented signal scenarios. Results indicated that the performance of our proposed model was more robust compared and provided relatively



(a) example of complete degradation signals



(b) prediction error for functional regression



(c) prediction error for exp-Brown

Figure 4.6: Plots of prediction errors under complete signals scenario.

failure predictability in comparison to the other benchmarks used in the study. This was particularly true to for sparse and fragmented degradation signals. In the case of complete

signals that had no missing data, our model performance at least as good as the benchmark parametric model.

The model proposed in this chapter is limited to applications with a single failure mode. However, we believe it is possible to extend this modeling framework to encompass multiple failure modes. From the sensing perspective, the model used a single type of sensor observation. This is also a limitation from a practical standpoint as often time multiple sensors are being used to monitor a single component. Consequently, future research is still needed to account for this aspect

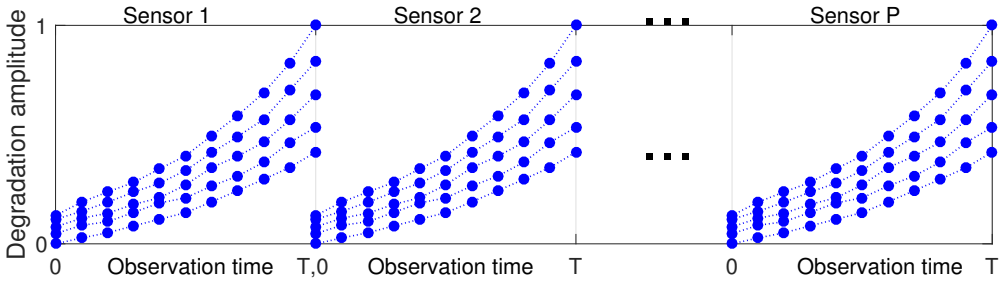
CHAPTER 5
MULTI-SENSOR PROGNOSTIC MODELING FOR APPLICATIONS WITH HIGHLY INCOMPLETE SIGNALS: A MATRIX COMPLETION APPROACH

5.1 Introduction

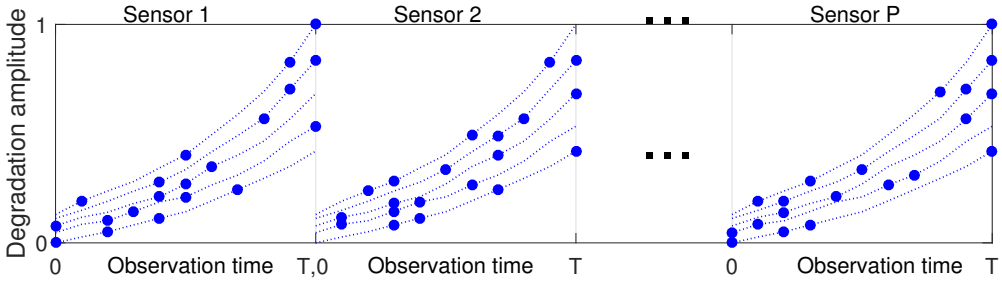
Inexpensive sensor technology has allowed many original equipment manufacturers to install numerous sensors on their products, especially capital-intensive assets. These sensors are used to detect faults and determine the severity of an asset's degradation state through condition monitoring. Prognostics is the process of transforming raw condition monitoring data into high-fidelity degradation signals to predict the residual lifetime of an asset. However, many of these complex assets operate in harsh environments that often have a significant impact on the quality of the raw data due to errors in data acquisition, communication, read/write operations, etc. Consequently, the resulting degradation signals often contain significant levels of missing and corrupt observations (aka. *incomplete signals*). This chapter focuses on multi-sensor prognostics of capital-intensive assets with highly incomplete degradation signals. One of the key contributions of this chapter is the development of a prognostic methodology capable of modeling poor quality multi-stream degradation signals to predict residual lifetime. A secondary contribution is incorporating parallelizable algorithms that leverage the low rank structure of our problem settings to enable the scalability of our methodology to Big Data settings involving large numbers of complex assets.

There have been several types of multi-stream prognostic models proposed in the literature, such as neural network models [22], neuro-fuzzy methods [28], parametric models that utilize data aggregation and fusion methods [32], and functional principal component analysis [97]. All of these models have been developed on the premise that the degradation

signals are observed *with high fidelity at frequent time steps*. In reality, however, degradation observations often contain outliers as well as missing and corrupt data which we define as *incomplete*. Figure 5.1(a) shows examples of multi-stream degradation signals with no poor quality aspects. In contrast, Figure 5.1(b) shows an example of the *incomplete* case. In this chapter, we use the term *highly incomplete* to define settings where at least 50% of signal observations are either missing or corrupt.



(a) Complete (regular observation grids)



(b) Incomplete

Figure 5.1: An illustration of complete and incomplete degradation signals.

Several key papers have investigated degradation modeling in the context of missing observations [63, 64, 19, 98, 99]. The methodologies developed in almost all these papers were based on functional principal component analysis (FPCA) and kernel smoothers. Specifically, FPCA was used to extract signal features (known as FPC-scores), which were then estimated from incomplete degradation signals via kernel smoothers. For example, papers [63, 64, 19] focused on developing prognostic models for single sensor applications with incomplete degradation signals, especially fragmented and sparsely observed signals. Papers [98, 99] proposed algorithms to recovery missed observations from observed in-

complete signals.

One common limitation of the research mentioned above is that they assume the degradation signals share the same time domain. This is because all these models used FPCA, which requires each observed curve shares the same time domain. In reality, however, this is not true due to signal *truncation*. That is, a system is usually stopped for repair or replacement when its degradation signal crosses a predefined "failure threshold", and thus no further observation can be acquired beyond that point. In such scenarios, using FPCA results in a significantly biased estimate of the mean and covariance functions due to the fact unobserved data beyond the threshold [64]. To address this challenge, the authors in [64] proposed a procedure that relied on axis transformation. Specifically, instead of plotting the degradation level on the y-axis with time on the x-axis, they reversed the axes. However, this approach was limited to strictly monotonic signals with very low noise levels. Paper [19] proposed a time-varying regression framework to address the time-domain issue. The basis of the time-varying regression is that training systems whose lifetime is smaller than the current observation time of the test signals are removed from the training dataset. And the chosen training signals are then truncated at the current observation time. However, the time-varying structure in [19] is computationally expensive since it generates new training dataset, and thus the model has to be re-estimated each time a new observation is observed from the fielded system. In addition, it does not make full use of the available dataset. To address the time-domain challenge, in this chapter, we propose a signal transformation framework based on time-domain to polar-domain transformation. This transformation enables us to model degradation signals with different lifetime and hence different lengths.

Another common limitation for all the aforementioned models is that they are computationally expensive. The computational burden mainly results from the kernel smoothers, which are used to estimate the signal features extracted from FPCA. Specifically, a one-dimensional kernel smoother is used to estimate the mean function, and a two-dimensional kernel smoother is utilized to smooth the covariance function. It is well known that ker-

nel smoothers are compute-intensive [48], especially for large-scale signal/covariance matrices. To reduce the computational burden, the estimation algorithms developed in this chapter are highly efficient on computation. More details will be discussed later.

The third common limitation for the research mentioned above is that almost all of them were designed for applications where equipment are monitored using a *single sensor*. Two exceptions are [98, 99], in which the authors used functional regression to recovery missed signal observations of one sensor by using signals from another sensor. However, the methodology can only simultaneously use signals from at most two sensors. In this chapter, we propose a prognostics model for *multi-sensor* applications where the multi-stream degradation signals are highly incomplete. The model is based on functional LLS regression in which the predictor is a set of multi-stream degradation signals and the response is TTF. The estimation of a functional LLS regression is usually an intractable problem. As a result, we first use multivariate FPCA to fuse the multi-stream signals. This enables us to transform the functional regression framework to a classic LLS regression model, in which the predictor is the fused features (known as "FPC-scores") from multivariate FPCA and the response is TTF.

Multivariate FPCA is capable of capturing the joint variation of multi-stream functional data (degradation signals in our case). To estimate the FPC-scores, all the existing estimation methods assume that signals are complete, that is, they are observed continuously and frequently at regular time grids. To be specific, the complete signals from different sensors are first concatenated to form a signal matrix. Next, SVD is applied to the signal matrix (or equivalently, Eigen Decomposition, ED, is applied to the covariance matrix of the signal matrix) to compute singular (eigen) vectors. Finally, FPC-scores are estimated by projecting signals to the singular vectors. For incomplete signals, however, none of the existing estimation method can work. This is because when the signal matrix is incomplete, neither SVD nor ED can be used to compute the singular vectors. To address this challenge, two algorithms are developed in this chapter. The first algorithm, called *subspace detection*, first

extracts a basis of the subspace that the degradation signals lie in, by utilizing the incomplete observations. Next, with the help of the basis, a novel feature extraction algorithm is developed to compute the singular vectors of the signal matrix. Finally, FPC-scores are calculated using the singular vectors and the incomplete signals. The second algorithm, referred to as *signal recovery*, begins with recovering the degradation signals from each sensor via its incomplete observations. Next, the recovered signals from different sensors are concatenated. To address the computational challenge when the concatenated signal matrix is big, we develop an incremental SVD algorithm, which computes the singular vectors of the concatenated signal matrix by adding one of its columns at a time. Finally, FPC-scores are computed using the incomplete signals and the singular vectors.

The remainder of the chapter is organized as follows. In Section 5.2 we present the degradation modeling and prognostics framework. We then discuss the subspace detection algorithm in Section 5.3 and the signal recovery algorithm in Section 5.4. The performance of our model is evaluated using simulated data in Section 5.5 and aircraft turbofan engine degradation data in Section 5.6. Finally, Section 5.7 concludes.

5.2 Degradation modeling and prognostics framework

This chapter focuses on developing a prognostic model for systems that are monitored by multiple sensors. We assume that data from each sensor is synthesized into one type of degradation signal. The prognostic model is established using functional LLS regression, in which the covariate is the multi-stream degradation signals and the response is TTF. In addition, we assume a historical dataset is available for model estimation. The historical dataset, also known as *training dataset*, contains degradation signals from a set of (identical) systems coupled with their corresponding TTFs. Consider a training dataset of N units and each unit is monitored by P sensors. For system i, denote its TTF as $\tilde{y}_i$ and its degradation signal from the pth sensor as $x_{i,p}(t)$, where $i = 1, \ldots, N, p = 1, \ldots, P$ and $t \in [0, T]$. We use the functional LLS regression to model the relationship between the

TTF and degradation signals:

$$y_i = \gamma_0 + \int_0^T \boldsymbol{\gamma}(t)^\top \boldsymbol{x}_i(t)dt + \sigma\epsilon_i \tag{5.1}$$

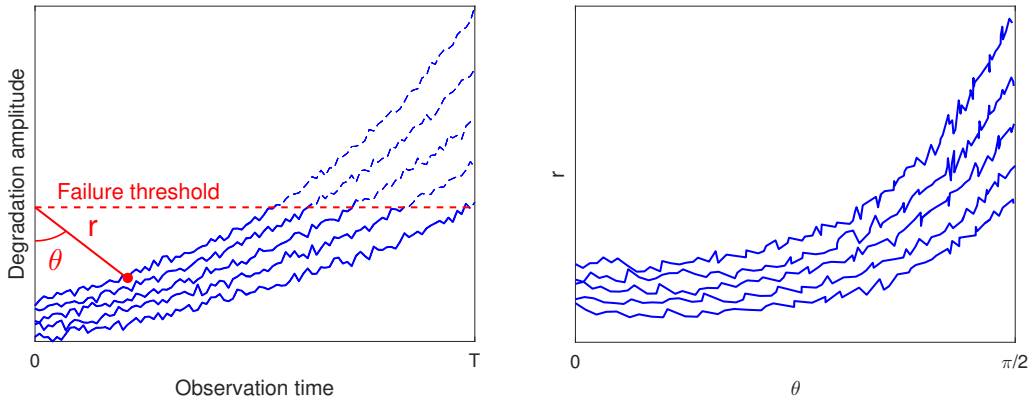
where $y_i = \tilde{y}_i$ for location-scale model and $y_i = \ln(\tilde{y}_i)$ for log-location-scale model, γ_0 is the intercept, $\boldsymbol{\gamma}(t) = (\gamma_1(t), \ldots, \gamma_P(t))^\top$ is the regression coefficient and $\boldsymbol{x}_i(t) = (x_{i,1}(t), \ldots, x_{i,P}(t))^\top$ is the concatenated signals from all P sensors, σ is the scale parameter and ϵ_i is the random noise term with a standard location-scale density $f(\epsilon)$. For example, $f(\epsilon) = \exp(\epsilon - \exp(\epsilon))$ for SEV distribution and $f(\epsilon) = 1/\sqrt{2\pi}\exp(-\epsilon^2/2)$ for normal distribution. Consequently, y_i has a density in the form of $\frac{1}{\sigma} f\left(\frac{y_i - \gamma_0 - \int_0^T \boldsymbol{\gamma}(t)^\top \boldsymbol{s}_i(t)dt\rangle}{\sigma}\right)$.

5.2.1 Polar-domain transformation of degradation signals

Functional LLS regression in Equation (5.1) requires that the degradation signals from each type of sensor share the same time domain. In reality, however, this is not true due to signal *truncation*. That is, a system is usually stopped for repair or replacement when its degradation signal crosses a predefined "failure threshold", and thus no further observation can be acquired beyond that point. As an illustration, Figure 5.2(a) shows degradation signals from five systems measured by one sensor. In the figure, only the solid portions can be observed. To address this challenge, we propose a polar coordinate transformation method. To be specific, we notice that the observable portion of all the degradation signals share the same polar domain, $[0, \pi/2]$ (see θ in Figure 5.2(a)). Therefore, we express degradation signals use their polar coordinates. Specifically, for observation $(t, s_{i,p}(t))$, let

$$\begin{cases} \theta = \arctan\left(\frac{t}{m_p - s_{i,p}(t)}\right) \\ r_{i,p}(\theta) = \sqrt{t^2 + (m_p - s_{i,p}(t))^2}, \end{cases} \tag{5.2}$$

where m_p is the failure threshold for sensor p. Then all the transformed signals from sensor p, i.e., $\{r_{i,p}(\theta)\}_{i=1}^N$, share the same domain $[0, \frac{\pi}{2}]$ (see Figure 5.2(b)). As a result, the



(a) Truncation of degradation signals (b) Transformed signals in polar coordinate

Figure 5.2: The truncation of degradation signals and polar coordinate transformation.

functional LLS regression model in Equation (5.1) can be expressed as follows:

$$y_i = \alpha_0 + \int_0^{\frac{\pi}{2}} \boldsymbol{\alpha}(\theta)^\top \boldsymbol{r}_i(\theta) d\theta + \sigma\epsilon_i \tag{5.3}$$

where α_0 is the intercept, $\boldsymbol{\alpha}(\theta) = (\alpha_1(\theta), \ldots, \alpha_P(\theta))^\top$ is the regression coefficient and $\boldsymbol{r}_i(\theta) = (r_{i,1}(\theta), \ldots, r_{i,P}(\theta))^\top$ is the transformed degradation signals for system i.

5.2.2 Multi-stream degradation signal fusion

The estimation of a functional LLS regression model is nontrivial. To address this challenge, multivariate FPCA is employed to fuse the multi-stream signals. Multivariate FPCA is an extension of FPCA. It works by concatenating different types of degradation signals into a single vector, and FPCA is then applied to the concatenated vector in a conventional manner. One benefit of multivariate FPCA is that it is capable of capturing the auto- and cross-correlation within/among signal streams and providing low-dimensional fused features. More importantly, it can be proven that, by incorporating multivariate FPCA, the functional LLS regression model can be equivalently transformed to a classic LLS regression model, where the covariate is the fused features and the response is still TTF (see Appendix B for more details). The classic LLS regression can then be estimated by utiliz-

ing maximum likelihood estimation.

Let the mean and covariance function of the degradation signals (i.e., $\{\boldsymbol{r}_i(\theta)\}_{i=1}^N$) be $\boldsymbol{\mu}(\theta) = (\mu_1(\theta), \ldots, \mu_P(\theta))^\top$ and $\boldsymbol{C}(\theta, \theta')$, respectively. Then $\boldsymbol{C}(\theta, \theta')$ is a $P \times P$ block matrix, where the (g, h)th block is the covariance function between sensor g and h, for $g = 1, \ldots, P$ and $h = 1, \ldots, P$, with $\theta, \theta' \in [0, \frac{\pi}{2}]$. Using Mercer's theorem, $\boldsymbol{C}(\theta, \theta')$ can be decomposed as $\boldsymbol{C}(\theta, \theta') = \sum_{k=1}^{\infty} \eta_k \boldsymbol{\psi}_k(\theta) \boldsymbol{\psi}_k(\theta')^\top$, where $\eta_1 \geq \eta_2 \geq \ldots$, are eigenvalues, and $\boldsymbol{\psi}_k(\theta) = (\psi_{k,1}(\theta), \ldots, \psi_{k,P}(\theta))^\top$ for $k = 1, 2, \ldots$ are the corresponding eigenfunctions. Thus, we can rewrite $\boldsymbol{r}_i(\theta)$ as follows:

$$\boldsymbol{r}_i(\theta) = \boldsymbol{\mu}(\theta) + \sum_{k=1}^{\infty} \zeta_{i,k} \boldsymbol{\psi}_k(\theta), \tag{5.4}$$

where $\zeta_{i,k} = \int_0^{\frac{\pi}{2}} (\boldsymbol{r}_i(\theta) - \boldsymbol{\mu}(\theta))^\top \boldsymbol{\psi}_k(\theta) d\theta$ are the FPC-scores. It is often sufficient to use a few eigenfunctions corresponding to the largest eigenvalues to approximate signals with a reasonable accuracy. Using only K eigenfunctions, equation (5.4) can now be rewritten as $\boldsymbol{r}_i(\theta) = \boldsymbol{\mu}(\theta) + \sum_{k=1}^{K} \zeta_{i,k} \boldsymbol{\psi}_k(\theta)$. Since the set of eigenfunctions $\{\boldsymbol{\psi}_k(\theta)\}_{k=1}^{\infty}$ forms a complete orthonormal basis, $\boldsymbol{\alpha}(\theta)$ can be expanded to $\boldsymbol{\alpha}(\theta) = \sum_{k=1}^{\infty} \beta_k \boldsymbol{\psi}_k(\theta)$. Therefore, the functional LLS model in Equation (5.3) can be expressed as follows (details of the derivation can be found in [97]):

$$y_i = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{\zeta}_i + \sigma \epsilon_i, \tag{5.5}$$

where β_0 is the intercept, $\boldsymbol{\beta} \in \mathbb{R}^{K \times 1}$ is the coefficient and $\boldsymbol{\zeta}_i = (\zeta_{i,1}, \ldots, \zeta_{i,K})^\top \in \mathbb{R}^{K \times 1}$ is the FPC-scores for system i, which can be estimated from historical degradation signals. Given $\boldsymbol{\zeta}_i$, the parameters in Equation (5.5), i.e., $(\beta_0, \boldsymbol{\beta}, \sigma)$, can be estimated using maximum likelihood estimation.

5.2.3 Model estimation with complete signals

In this subsection, we discuss how to estimate the FPC-scores when the signals are complete. Denote the discrete observation time point for sensor p as $\{\Theta_{p,1}, \Theta_{p,2}, \ldots, \Theta_{p,J_p}\}$, where J_p is the number of observations for sensor p. Then, the discrete observations for sensor p of system i are $\boldsymbol{l}_{i,p} = (\boldsymbol{r}_{i,p}(\Theta_{p,1}), \boldsymbol{r}_{i,p}(\Theta_{p,2}), \ldots, \boldsymbol{r}_{i,p}(\Theta_{p,J_p}))^\top \in \mathbb{R}^{J_p \times 1}$. Thus, the concatenated signals from all the P sensors of system i is $\boldsymbol{s}_i = (\boldsymbol{l}_{i,1}, \boldsymbol{l}_{i,2}, \ldots, \boldsymbol{l}_{i,P})^\top \in \mathbb{R}^{M \times 1}$, where $M = \sum_{p=1}^{P} J_p$. Then, the observed degradation signal matrix from all the P sensors and N systems is $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N) \in \mathbb{R}^{M \times N}$. Without loss of generality, we assume $N < M$. The FPC-scores, $\boldsymbol{\zeta}_i$, in Equation (5.5) can be estimated as follows:

(i) Estimating signal mean. The signal mean is computed by taking the signal average over systems, i.e., $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{s}_i$, where $\boldsymbol{s}_i$ is the ith column of $\boldsymbol{S}$.

For sensor p, its signal mean can be computed by taking the average over systems, i.e., $\hat{\boldsymbol{\mu}}_p = \left(\frac{1}{N}\sum_{i=1}^{N} r_{i,p}(\Theta_{p,1}), \frac{1}{N}\sum_{i=1}^{N} r_{i,p}(\Theta_{p,2}), \ldots, \frac{1}{N}\sum_{i=1}^{N} r_{i,p}(\Theta_{p,J_p})\right)^\top$. Then, the estimated mean function is $\hat{\boldsymbol{\mu}} = (\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \ldots, \hat{\boldsymbol{\mu}}_P)^\top$

(ii) Centralizing signal matrix. This is done by setting $\tilde{\boldsymbol{s}}_i = \boldsymbol{s}_i - \hat{\boldsymbol{\mu}}$, for $i = 1, \ldots, N$, and $\tilde{\boldsymbol{S}} = (\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \ldots, \tilde{\boldsymbol{s}}_N)$ is the centralized signal matrix.

(iii) Estimating eigenvectors. Eigenvectors are estimated by solving the eigen equation $\tilde{\boldsymbol{S}}\boldsymbol{\psi} = \eta\boldsymbol{\psi}$, which can be done by applying SVD on the matrix $\tilde{\boldsymbol{S}}$. The resulting eigenvalues associated with the eigenvectors are denoted as $\{\hat{\eta}_k, \hat{\boldsymbol{\psi}}_k\}$ for $k = 1, \ldots, N$. Select the first K eigenvectors by using fraction-of-variance explained (FVE) as follows: $K = \inf_k\{F_k \geq D\}$, where $F_k = \sum_{j=1}^{k} \eta_j^2 / \sum_{j=1}^{N} \eta_j^2$ and $D \in (0, 1]$ is the FVE threshold.

(iv) Computing the FPC-scores. The kth FPC-score of system i is $\zeta_{i,k} = \tilde{\boldsymbol{s}}_i^\top \hat{\boldsymbol{\psi}}_k$, for $k = 1, \ldots, K$ and $\boldsymbol{\zeta}_i = (\zeta_{i,1}, \zeta_{i,2}, \ldots, \zeta_{i,K})^\top$.

It can be seen that none of the above steps can work when data is incomplete. To address this challenge, we propose two estimation methodologies, i.e., *subspace detection* and *signal recovery*, both of which will be explained in the following sections.

5.3 Fusing highly incomplete signals using subspace detection

In this section, we propose an algorithm called *subspace detection* to estimate the FPC-scores in Equation (5.5) by utilizing highly incomplete degradation signals. Specifically, we first detect the subspace of the signal matrix using its incomplete observations, and extract an orthonormal basis that spans the subspace. Next, with the help of the basis, we develop a method that computes the SVD of the centered signal matrix via its incomplete observations. Finally, the singular vectors resulting from SVD are used to compute the FPC-scores.

$$(\boldsymbol{s}_i^{\Omega_i})_j = \begin{cases} (\boldsymbol{s}_i)_j, & \text{if } \ j \in \Omega_i \\ 0, & \text{otherwise} \end{cases} . \tag{5.6}$$

Denote the discrete observation time point for sensor p as $\{\Theta_{p,1}, \Theta_{p,2}, \ldots, \Theta_{p,J_p}\}$, where J_p is the number of observations for sensor p. Then, the discrete observations from sensor p of system i are $\boldsymbol{l}_{i,p} = (\boldsymbol{r}_{i,p}(\Theta_{p,1}), \boldsymbol{r}_{i,p}(\Theta_{p,2}), \ldots, \boldsymbol{r}_{i,p}(\Theta_{p,J_p}))^\top \in \mathbb{R}^{J_p \times 1}$. And the concatenated signals from all the P sensors of system i is $\boldsymbol{s}_i = (\boldsymbol{l}_{i,1}, \boldsymbol{l}_{i,2}, \ldots, \boldsymbol{l}_{i,P})^\top \in \mathbb{R}^{M \times 1}$, where $M = \sum_{p=1}^{P} J_p$. Thus, the observed degradation signal matrix from all the P sensors and N systems is $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N) \in \mathbb{R}^{M \times N}$. Suppose the ith column, $\boldsymbol{s}_i$, are observed at locations $\Omega_i \subseteq \{1, 2, \ldots, M\}$. Let $\boldsymbol{s}_i^{\Omega_i}$ be the $M \times 1$ vector contains the revealed entries of $\boldsymbol{s}_i$, and is null in other positions. That is, $(\boldsymbol{s}_i^{\Omega_i})_j = (\boldsymbol{s}_i)_j$ if $j \in \Omega_i$ and null otherwise. Then $\boldsymbol{\Omega} = \{(j, i) : j \in \Omega_i, i = 1, \ldots, N\}$ contains the index of observed entries of $\boldsymbol{S}$, and $\boldsymbol{S}^{\boldsymbol{\Omega}} = (\boldsymbol{s}_1^{\Omega_1}, \boldsymbol{s}_2^{\Omega_2}, \ldots, \boldsymbol{s}_N^{\Omega_N})$ represents the $M \times N$ matrix that contains the revealed entries of $\boldsymbol{S}$ and is null in other positions. Suppose the rank of the signal matrix $\boldsymbol{S}$ is K, which is unknown and will be estimated later. Let $\boldsymbol{B} \in \mathbb{R}^{M \times K}$ be any matrix whose columns span the range (column space) of $\boldsymbol{S}$. Matrix $\boldsymbol{S}$ can be decomposed as $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{K \times N}$ is the weight matrix. Given the incomplete observed signal matrix $\boldsymbol{S}^{\boldsymbol{\Omega}}$, $\boldsymbol{B}$ can

be estimated by optimizing the following problem [100, 101]:

$$\min_{\boldsymbol{B},\boldsymbol{A}} \|\boldsymbol{S}^{\boldsymbol{\Omega}} - (\boldsymbol{B}\boldsymbol{A})^{\boldsymbol{\Omega}}\|_F^2 + \lambda(\|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2), \tag{5.7}$$

where $\|\cdot\|_F^2$ is the Frobenius norm and λ is the tuning parameter. Optimization problem (5.7) can be solved by many existing algorithms [102]. Solving optimization problem (5.7) provides a basis matrix $\boldsymbol{B}$ for the range of $\boldsymbol{S}$. Then we can orthonormalize matrix $\boldsymbol{B}$ to get an orthonormal basis (denoted as $\boldsymbol{Q}$ hereafter) for the range of $\boldsymbol{S}$.

Next, with the help of $\boldsymbol{Q}$, we develop a method that computes the SVD of centered $\boldsymbol{S}$ (denoted by $\tilde{\boldsymbol{S}}$) by using the incomplete observations of $\boldsymbol{S}$. To do this, we first express the signal matrix $\boldsymbol{S}$ using basis $\boldsymbol{Q}$. In particular, the ith column of $\boldsymbol{S}$ can be expressed as a linear combination of the columns of $\boldsymbol{Q}$, i.e., $\boldsymbol{s}_i = \boldsymbol{Q}\boldsymbol{w}_i$, where $\boldsymbol{w}_i \in \mathbb{R}^{K\times 1}$ is the weight vector, for $i = 1, \ldots, N$ and $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N) \in \mathbb{R}^{K\times N}$ is the weight matrix. If the complete matrix $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N)$ is known, the weight vector $\boldsymbol{w}_i$ can be computed via $\boldsymbol{w}_i = \boldsymbol{Q}^\top \boldsymbol{s}_i$. However, the complete matrix is not available, and thus we have to relay on the incomplete matrix $\boldsymbol{S}^{\boldsymbol{\Omega}} = (\boldsymbol{s}_1^{\Omega_1}, \boldsymbol{s}_2^{\Omega_2}, \ldots, \boldsymbol{s}_N^{\Omega_N})$. Based on the available observations, the weight vector can be estimated as follows:

$$\boldsymbol{w}_i = \min_{\boldsymbol{w}_i} \|\boldsymbol{Q}^{\Omega_i}\boldsymbol{w}_i - \boldsymbol{s}_i^{\Omega_i}\|^2, \tag{5.8}$$

where matrix $\boldsymbol{Q}^{\Omega_i} \in \mathbb{R}^{|\Omega_i|\times K}$ consists the $|\Omega_i|$ rows of matrix $\boldsymbol{Q}$ indexed by the set Ω_i. After estimating the weight vectors, we centralize the weight matrix $\boldsymbol{W}$. Specifically, let the column mean be $\overline{\boldsymbol{w}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{w}_i$ and the mean matrix $\overline{\boldsymbol{W}}$ consists K such mean vectors, i.e., $\overline{\boldsymbol{W}} = (\overline{\boldsymbol{w}}, \overline{\boldsymbol{w}}, \ldots, \overline{\boldsymbol{w}})$. As a result, the centered weight matrix is $\tilde{\boldsymbol{W}} = \boldsymbol{W} - \overline{\boldsymbol{W}}$. Finally, SVD is applied on $\tilde{\boldsymbol{W}}$, i.e., $\tilde{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{K\times K}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{K\times N}$ and $\boldsymbol{V} \in \mathbb{R}^{N\times N}$. The SVD of the centered signal matrix can be expressed as $\tilde{\boldsymbol{S}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\top$, where $\hat{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{U}, \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}, \hat{\boldsymbol{V}} = \boldsymbol{V}$.

We summarize the aforementioned procedure in Algorithm 3. Proposition 1 shows that

Algorithm 3 indeed computes the SVD of the centered matrix $\tilde{\boldsymbol{S}}$. The proof of Proposition 1 can be found in the appendix.

Algorithm 1: SVD decomposition for incomplete matrix

Input : An orthonormal basis of matrix $\boldsymbol{S}$ and its incomplete observations $\boldsymbol{S}^{\boldsymbol{\Omega}}$

Output: SVD decomposition of the centered matrix $\tilde{\boldsymbol{S}}$, i.e., $\tilde{\boldsymbol{S}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\top$

1 Computing the weight matrix:
$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N)$, where $\boldsymbol{w}_i = \min_{\boldsymbol{w}_i} \|\boldsymbol{Q}^{\Omega_i}\boldsymbol{w}_i - \boldsymbol{s}_i^{\Omega_i}\|^2$.

2 Centralizing the weight matrix:
$\tilde{\boldsymbol{W}} = \boldsymbol{W} - \overline{\boldsymbol{W}}$, where $\overline{\boldsymbol{W}} = (\overline{\boldsymbol{w}}, \overline{\boldsymbol{w}}, \ldots, \overline{\boldsymbol{w}})$ and $\overline{\boldsymbol{w}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{w}_i$.

3 Applying SVD on the centered weight matrix:
$\tilde{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$

4 Setting $\hat{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{U}, \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}, \hat{\boldsymbol{V}} = \boldsymbol{V}$

Proposition 2 *Given an orthonormal basis (denoted by $\boldsymbol{Q}$) of the uncentered matrix $\boldsymbol{S}$ and its incomplete observations $\boldsymbol{S}^{\boldsymbol{\Omega}}$, Algorithm 3 computes the SVD of the centered matrix $\tilde{\boldsymbol{S}}$.*

Algorithm 3 provides the singular vectors of the signal matrix $\boldsymbol{S}$. By using the singular vectors, we can compute the FPC-scores of the signals from each system (i.e., each column of $\boldsymbol{S}$). If $\boldsymbol{S}$ is complete, the FPC-scores of system i, i.e., $\boldsymbol{\zeta}_i \in \mathbb{R}^{K\times 1}$, can be estimated via $\boldsymbol{\zeta}_i = \hat{\boldsymbol{U}}^\top \boldsymbol{s}_i$. For incomplete data, it can be estimated as follows:

$$\boldsymbol{\zeta}_i = \min_{\boldsymbol{\zeta}_i} \|\hat{\boldsymbol{U}}^{\Omega_i}\boldsymbol{\zeta}_i - \boldsymbol{s}_i^{\Omega_i}\|^2, \tag{5.9}$$

where matrix $\hat{\boldsymbol{U}}^{\Omega_i} \in \mathbb{R}^{|\Omega_i|\times K}$ consists the $|\Omega_i|$ rows of matrix $\hat{\boldsymbol{U}}$ indexed by the set Ω_i.

5.4 Fusing highly incomplete signals using signal recovery

In this section, we propose another algorithm to estimate the FPC-scores in Equation (5.5) by utilizing the high incomplete multi-stream degradation signals. Specifically, we first use matrix completion techniques to recover the signals from each sensor. Next, the recovered signals from different sensors are concatenated and SVD is applied to it to extract the singular vectors. To address the computational challenge of traditional SVD when

the recovered signal matrix is with large size, we develop an incremental SVD algorithm, which computes the SVD by adding one column of the signal matrix at a time. Finally, the FPC-scores are calculated using the singular vectors and the observed incomplete signals.

Let $\boldsymbol{G}_p \in \mathbb{R}^{J_p \times N}$ be the degradation signal matrix from the pth sensor, where J_p is the number of observation time point for sensor p and N is the number of system. Out of the $J_p \times N$ entries of $\boldsymbol{G}_p$, a subset $\boldsymbol{\Omega} \subseteq \{(j,i) : j = 1, \ldots, J_p, i = 1, \ldots, N\}$ are observed. Define the projector operator $\mathcal{P}_{\boldsymbol{\Omega}}(\cdot)$ as follows:

$$[\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{G}_p)]_{j,i} = \begin{cases} (\boldsymbol{G}_p)_{j,i}, & \text{if } (j,i) \in \boldsymbol{\Omega} \\ 0, & \text{otherwise} \end{cases} . \tag{5.10}$$

To recover the signals of sensor p by using its incomplete observations, we need to find a matrix with the minimum rank that best matches the observations. This is known as a matrix completion problem [103]. Consider the fact that the degradation signals are contaminated with noise, we optimize the following criterion

$$\begin{aligned} \min \quad & \|\boldsymbol{X}\|_* \\ \text{subject to} \quad & \|\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{X}) - \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{G}_p)\|_F^2 \leq \delta \end{aligned} \tag{5.11}$$

where $\|\cdot\|_*$ is the nuclear norm, $\|\cdot\|_F^2$ is the Frobenius norm and $\delta > 0$ is some constant. Many efficient algorithms can be used to solve (6.3). Such as the Accelerated Proximal Gradient [102]. Solving (6.3) provides the recovered signal matrix from sensor p, i.e., $\hat{\boldsymbol{G}}_p = \boldsymbol{X}$.

After recovering the signals from each sensor, the signal matrix from all the sensors can be built as follows: $\hat{\boldsymbol{S}} = (\hat{\boldsymbol{G}}_1^\top, \hat{\boldsymbol{G}}_2^\top, \ldots, \hat{\boldsymbol{G}}_P^\top)^\top$, which is with dimensionality of $M \times N$, where $M = \sum_{p=1}^{P} J_p$. Next, we apply SVD on the centered $\hat{\boldsymbol{S}}$, denoted by $\tilde{\boldsymbol{S}}$, to extract its singular vectors. However, SVD is computationally expensive if matrix $\tilde{\boldsymbol{S}}$ is with large size. To address this challenge, we use incremental SVD, which computes the SVD by adding one column of the signal matrix at a time. The existing incremental SVD algorithms are de-

signed for matrices with full-rank or low-rank. In our case, however, the degradation signal matrix is approximately low-rank, which implies that some of its singular values are very close to but not exactly zeros. These small singular values represent the variation resulting from signal noise. For matrices with approximately low-rank, none of the existing incremental SVD algorithms can work. To address this challenge, in this chapter, we develop an incremental SVD algorithm that can work with matrices with approximately low-rank. The algorithm is shown in Algorithm 2. In the algorithm, to be consistent with the notation in the former sections, we let $\tilde{\boldsymbol{s}}_i \in \mathbb{R}^{M\times 1}$ represent the ith column of $\tilde{\boldsymbol{S}}$. Using Algorithm 2, we can compute the SVD of the centered matrix $\tilde{\boldsymbol{S}}$ and the singular vector matrices $\boldsymbol{U}$, which can then be used to compute the FPC-scores via $\boldsymbol{\zeta}_i = \boldsymbol{U}^\top \tilde{\boldsymbol{s}}_i$.

Algorithm 2: Incremental SVD for matrix with approximately low-rank

Input : Matrix $\tilde{\boldsymbol{S}}_{[M\times N]}$, the subscript in $_{[\,]}$ is the dimensionality of the matrix, $\varepsilon = 1 \times e^{-6}$

Output: SVD of $\tilde{\boldsymbol{S}}$, i.e., $\tilde{\boldsymbol{S}}_{[M\times N]} = \boldsymbol{U}_{[M\times M]}\boldsymbol{\Sigma}_{[M\times N]}\boldsymbol{V}^\top_{[N\times N]}$

Initialization: $d = 1,\ \boldsymbol{U}_{[M\times d]} \coloneqq \frac{\tilde{\boldsymbol{s}}_1}{\|\tilde{\boldsymbol{s}}_1\|},\ \boldsymbol{\Sigma}_{[d\times d]} \coloneqq \|\tilde{\boldsymbol{s}}_1\|,\ \boldsymbol{V}_{[d\times d]} \coloneqq 1$

for $i = 2$ *to* N **do**

- $\boldsymbol{w}_{[d\times 1]} \coloneqq \boldsymbol{U}^\top_{[M\times d]}\tilde{\boldsymbol{s}}_{i[M\times 1]}$ % weight vector
- $\boldsymbol{p}_{[M\times 1]} \coloneqq \boldsymbol{U}_{[M\times d]}\boldsymbol{w}_{[d\times 1]}$
- $\boldsymbol{e}_{[M\times 1]} \coloneqq \tilde{\boldsymbol{s}}_{i[M\times 1]} - \boldsymbol{p}_{[M\times 1]}$ % residual vector
- $\begin{bmatrix}\boldsymbol{\Sigma}_{[d\times d]} & \boldsymbol{w}_{[d\times 1]} \\ \boldsymbol{0}_{[1\times d]} & \|\boldsymbol{e}\|_{[1\times 1]}\end{bmatrix} = \hat{\boldsymbol{U}}_{[(d+1)\times(d+1)]}\hat{\boldsymbol{\Sigma}}_{[(d+1)\times(d+1)]}\hat{\boldsymbol{V}}^\top_{[(d+1)\times(d+1)]}$
- $\boldsymbol{U}_{[M\times(d+1)]} \coloneqq \begin{bmatrix}\boldsymbol{U}_{[M\times d]} & \frac{\boldsymbol{e}_{[M\times 1]}}{\|\boldsymbol{e}\|_{[1\times 1]}}\end{bmatrix}\hat{\boldsymbol{U}}_{[(d+1)\times(d+1)]}$
- $\boldsymbol{\Sigma}_{[(d+1)\times(d+1)]} \coloneqq \hat{\boldsymbol{\Sigma}}_{[(d+1)\times(d+1)]}$
- $\boldsymbol{V}_{[(d+1)\times(d+1)]} \coloneqq \begin{bmatrix}\boldsymbol{V}_{[d\times d]} & \boldsymbol{0}_{[d\times 1]} \\ \boldsymbol{0}_{[1\times d]} & \boldsymbol{1}_{[1\times 1]}\end{bmatrix}\hat{\boldsymbol{V}}_{[(d+1)\times(d+1)]}$
- **if** $\|\boldsymbol{e}\| < \varepsilon$ **then**
  - $\boldsymbol{U}_{[M\times d]} \coloneqq \boldsymbol{U}_{[M\times(d+1)]}(1:M, 1:d)$ %delete the last column
  - $\boldsymbol{\Sigma}_{[d\times d]} \coloneqq \boldsymbol{\Sigma}_{[(d+1)\times(d+1)]}(1:d, 1:d)$ %delete both the last row and last column
  - $\boldsymbol{V}_{[(d+1)\times d]} \coloneqq \boldsymbol{V}_{[(d+1)\times(d+1)]}(1:d+1, 1:d)$ %delete the last column
- **else**
  - $d \coloneqq d + 1$
- **end**

end

5.5 Numerical study

In this section, we present a simulation study to validate the proposed prognostic model. We evaluate the performance of our approaches, designated as "Subspace detection" and "Signal recovery," in terms of the accuracy of predicting the RULs at different scenarios of data incompleteness.

We compare the performance of our methodologies to two benchmark models. The first benchmark, referred to as "Kernel smoother," is an extension of the prognostic model proposed by [37]. In [37], Hierarchical FPCA is utilized to fuse multi-stream degradation signals, and the fused features are then regressed against TTFs via LLS regression in a similar manner to the method proposed in this chapter. Hierarchical FPCA works by first applying FPCA to the degradation signals from each sensor (i.e., degradation signals are grouped by each sensor) individually to extract their FPC-scores. Next, the FPC-scores from different sensors are concatenated, and regular PCA is applied to the concatenated FPC-scores to extract fused features. The authors in [37] claimed that Hierarchical FPCA performed almost the same as multivariate FPCA (used in this chapter) on fusing multi-stream signals. However, the prognostic model proposed in [37] only works for complete data. Here, we extend it to incomplete data case by employing the kernel smoother algorithm developed in [48]. Specifically, kernel smoother is applied to the pooled data from all individuals of a same sensor to estimate its signal mean and covariance matrix. The covariance matrix is then decomposed using eigen decomposition and eigen vectors are provided. Using the eigen vectors and the incomplete observations, the FPC-scores can be computed via an algorithm called principal analysis by conditional expectation (PACE). More details about the algorithm can be found in [48]. Next, the FPC-scores from all the sensors are concatenated and regular PCA is applied on the concatenated vector to extract the fused features (i.e., $\boldsymbol{\zeta}_i$ in Equation (5.5)).

The second benchmarking model, designated "B Spline," is similar to the first bench-

mark except that the degradation signals are recovered using penalized B Spline. To be specific, penalized B Spline is first used to recovery the degradation signal of each sensor from each system individually. Next, the Hierarchical FPCA in [37] is applied to the recovered signals to extract features, which are then regressed against TTFs via LLS regression for RUL prediction. The tuning parameter of penalized B spline is selected by utilizing generalized cross validation (GCV) [104].

5.5.1 Data generation and validation settings

In this simulation study, we consider 200 identical systems, each of which is monitored by 10 sensors. We begin by simulating the degradation signals from the pth sensor of system i in the polar domain using the following functional form: $r_{i,p}(\theta) = 10 + c_{i,p}\sin(\theta)$, where $\{c_{i,p}\}_{i=1}^{10} \sim N(0, \sigma_p^2)$, $\{\sigma_p^2\} \sim N(1, 0.1)$. Next, we apply multivariate FPCA to the concatenated degradation signals to extract features. The number of FPC-scores is chosen by setting FVE at 0.95 (see Section 5.2.3 for details). As a result, the first 5 FPC-scores (i.e., $\hat{\boldsymbol{\xi}}_i = (\xi_{i,1}, \xi_{i,2}, \ldots, \xi_{i,5})^\top$) are chosen as the fused features of each system. The TTF, $\tilde{y}_i$, is then computed by using Equation (5.5), i.e., $\log(\tilde{y}_i) = \beta_0 + \boldsymbol{\beta}^\top \hat{\boldsymbol{\xi}}_i + \varepsilon_i$, where $\beta_0 = 2.2, \boldsymbol{\beta} = (0.01, 0.01, 0.01, 0.01, 0.01)^\top$ and $\varepsilon_i \sim N(0, 0.01)$.

$$s_{i,p}(t) = D - \cot\left(\arcsin\left(\frac{\sqrt{25 + c_{i,p}t} - 5}{c_{i,p}}\right)\right)t + \epsilon_i(t) \tag{5.12}$$

where $\epsilon_i(t) \sim N(0, 0.2), D = 10, t = [0 : 0.1 : TTF(i)]$.

We test the performance of our methodologies and the benchmarks using both complete and incomplete data. For incomplete data, we consider two sampling strategies: (1) *balance sampling* and (2) *imbalance sampling*. For balance sampling, all the sensors have the same level of incompleteness. Specifically, we consider four levels of data incompleteness: 20%, 40%, 60%, and 80%, where 20% means that we randomly select 20% observations from each signal. For imbalance sampling, different sensors have different level of data incompleteness. To be specific, we consider four different data incompleteness combina-

tions: "10%+90%,""20%+80%,""30%+70%," and "40%+60%." Here, "10%+90%" means that half of the sensors have 10% of their observations being randomly selected, and the other half have 90%. The simulation process is repeated 10 times. The prediction errors are computed using Equation (2.25). We summarize the mean absolute prediction errors at different life percentiles, where 10th represents the prediction errors evaluated at life percentiles in (5%; 15%], 20th represents the prediction errors evaluated at life percentiles in (15%; 25%], so on so forth.

5.5.2 Results and analysis

In this section, we report and analyze the prediction errors of our proposed methodology and the benchmarks at different levels of data incompleteness.

Complete signals

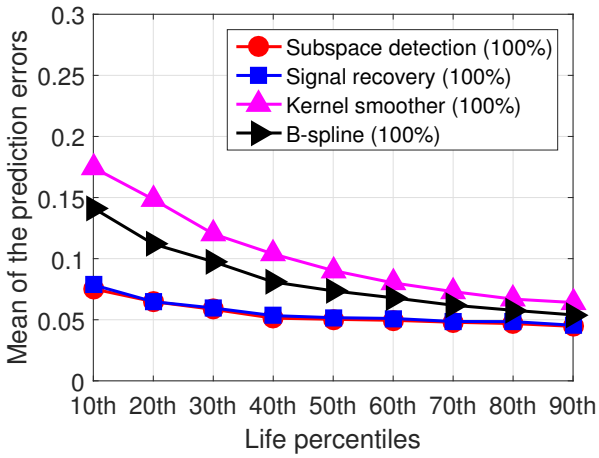


Figure 5.3: The mean prediction errors for complete signals.

Figure 5.3 presents the prediction errors when the signals are complete (i.e., containing no missing data). We observe that the prediction errors of "Subspace detection" and "Signal recovery" are similar. This observation implies that the two proposed estimation methodologies perform similarly when data contains no missing observations. In addition, Figure 5.3 indicates that our proposed methodologies perform better than both "Kernel smoother"

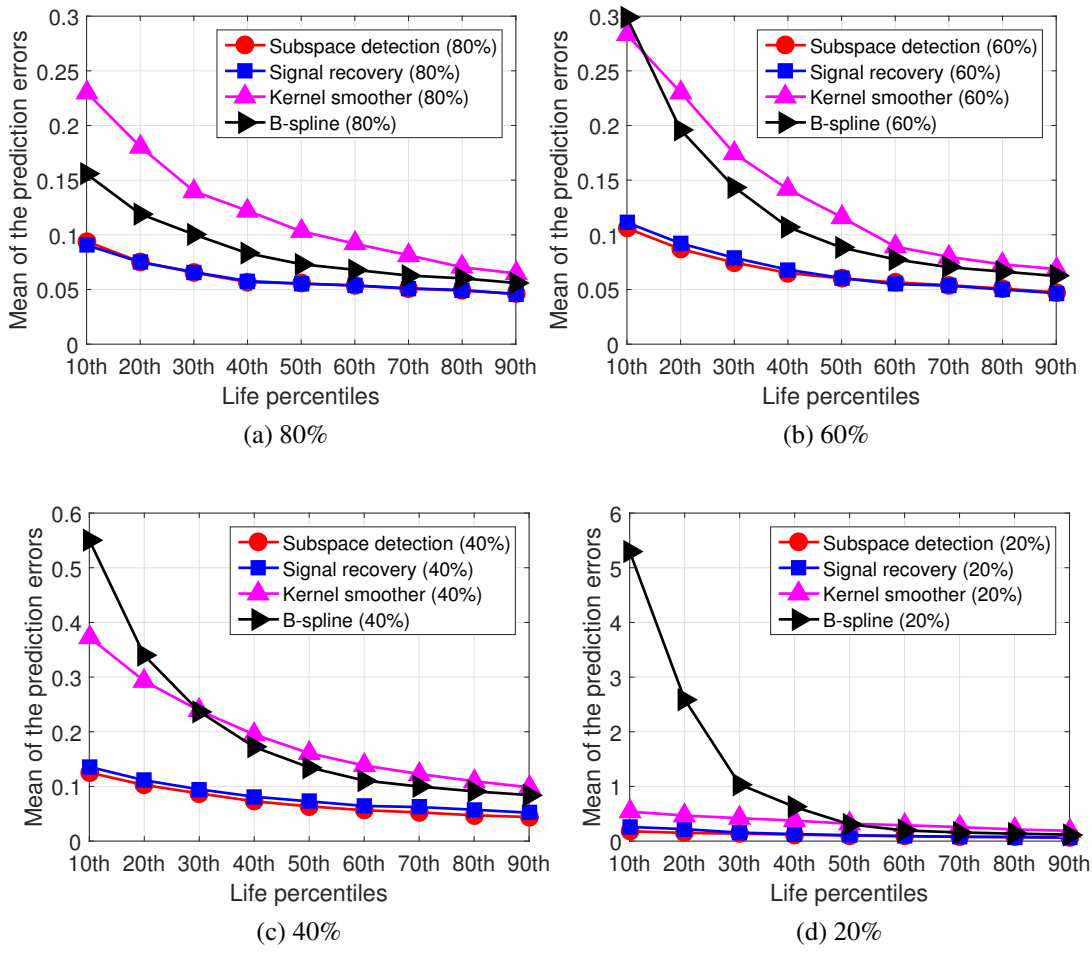
and "B Spline" consistently at all level of life percentiles. This is reasonable because our proposed methodologies use more sensor information to recovery signals. Specifically, to recovery a sensor signal of a specific system, say sensor p of system i, i.e., $s_{i,p}(t)$, the data used by "Subspace detection" includes: (i) the observations from $s_{i,p}(t)$ itself, (ii) the observations from sensor p of other training systems, i.e., $s_{\neq i,p}(t)$, and (iii) the observations from other sensors of both system i and other training systems, i.e., $s_{i,\neq p}(t)$ and $s_{\neq i,\neq p}(t)$, and the data used by "Signal recovery" consists of (i) and (ii). However, the only data used by "B Spline" is (i) since it fits each sensor signal individually. Although the other benchmark "Kernel smoother" also uses data (i) and (ii) as our proposed "Signal recovery," it only uses local data observations neighbored to the recovered one [19]. Therefore, both "B spline" and "Kernel smoother" perform worse than our proposed models.

Incomplete signals: Balance sampling

Figure 5.4 shows the prediction errors with balance sampled data. It can be observed that at all levels of data incompleteness, our proposed methodologies achieve smaller prediction errors than the benchmarking models. Figure 5.4 also indicates that "B spline" is sensitive to the data incompleteness level. For example, "B spline" performs much better than "Kernel smoother" when 80% signal observations are available (see Figure 5.4(a)). However, it performs similarly to the latter when the data availability is 40% (see Figure 5.4(c)), and its prediction errors are much larger than the latter when the data availability drops to 20% (see Figure 5.4(d)). This is reasonable since "B spline" only utilizes data (i) to recovery the signals, and thus the prediction accuracy is compromised when the available data is too limited.

Incomplete signals: Imbalance sampling

Figure 5.5 presents the prediction errors with imbalance sampled data. We can again observe that our proposed methodologies achieve smaller prediction errors than the bench-

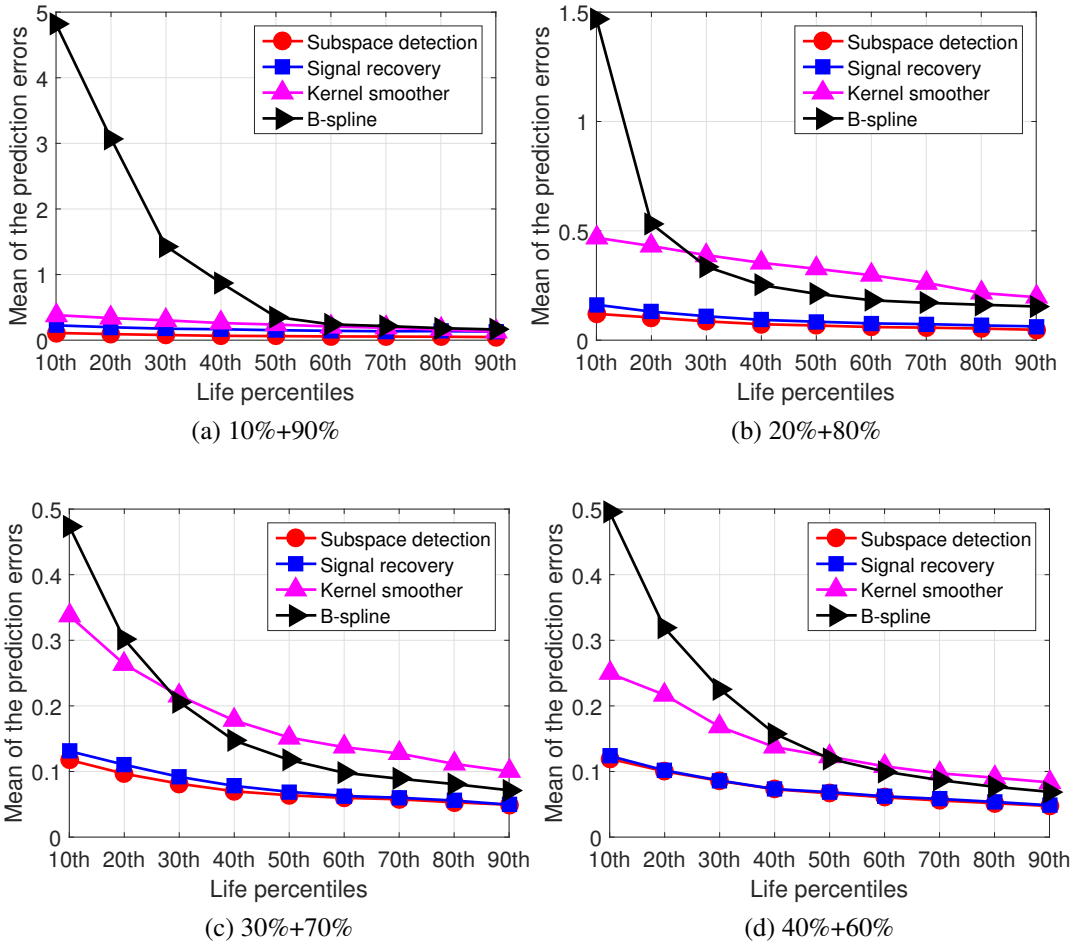


(a) 80% (b) 60%

(c) 40% (d) 20%

Figure 5.4: The mean prediction errors with balance sampled data.

marking models at all levels of data incompleteness. This confirms the benefit of using all data from (i), (ii), and (iii). Moreover, Figure 5.5 indicates that the prediction accuracy of "B spline" decreases significantly with the increase of data imbalance. For example, its 10th life percentile prediction errors at "40%+60%," "30%+70%," "20%+80%," and "10%+90%" imbalance combinations are around 0.5, 0.5, 1.5, and 5, respectively. Again, we believe this is because "B spline" only uses data (i). When data is highly imbalanced, the observed signal from some sensors are highly spare (such as Figure 5.5(a) where some sensors have only 10% data available). This results in a bad signal recovery accuracy for "B spline" and thus compromises the its prediction accuracy. In addition, Figure 5.5 also indicates that "Subspace detection" achieves better prediction accuracy than "Signal re-



(a) 10%+90%

(b) 20%+80%

(c) 30%+70%

(d) 40%+60%

Figure 5.5: The mean prediction errors with imbalance sampled data.

covery," especially when data is highly imbalanced. We believe this is because "Subspace detection" utilizes data (iii) while "Signal recovery" does not.

5.6 Case study

In this section, we use multi-sensor degradation data from aircraft turbofan engines provided by NASA [11] to evaluate the performance of our model. The dataset is comprised of the following; (i) degradation signals from 100 training engines that were run to failure, (ii) degradation signals from an additional 100 test engines whose operation was prematurely terminated at random time points prior to their failure time, and (iii) the real TTFs

of the 100 test engines. Each engine was monitored using 21 sensors. Following the suggestion of [97], we choose 4 sensors (i.e., Total temperature at LPT outlet, Bypass Ratio, Bleed Enthalpy and HPT coolant bleed) to build a prognostics model under lognormal distribution.

Both the training and test dataset are first transformed into polar coordinate system (discussed in Section 5.2). The transformed training dataset is then used for training and the transformed test dataset is used to validate the TTF prediction accuracy. Similar to the simulation study in Section 5.5, we use "Kernel smoother" and "B spline" to benchmark our proposed methodology and compare their performance using both complete and incomplete data. For incomplete data, we also consider two sampling strategies: (1) *balance sampling* and (2) *imbalance sampling*.

5.6.1 Results and analysis

In this section, we report and analyze the prediction errors of our proposed methodology and the benchmarks.

Complete signals

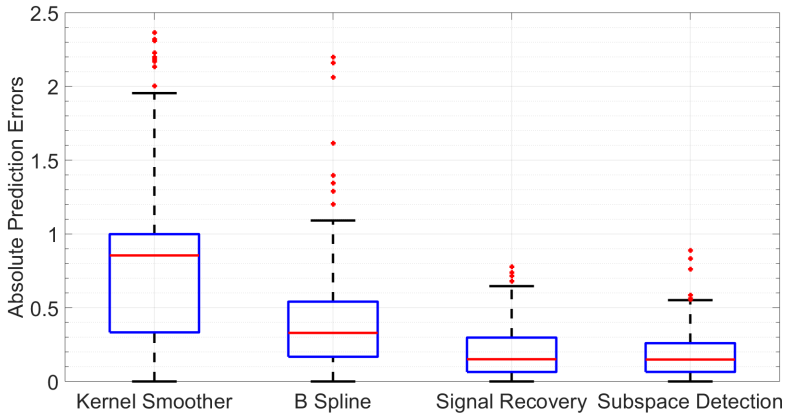
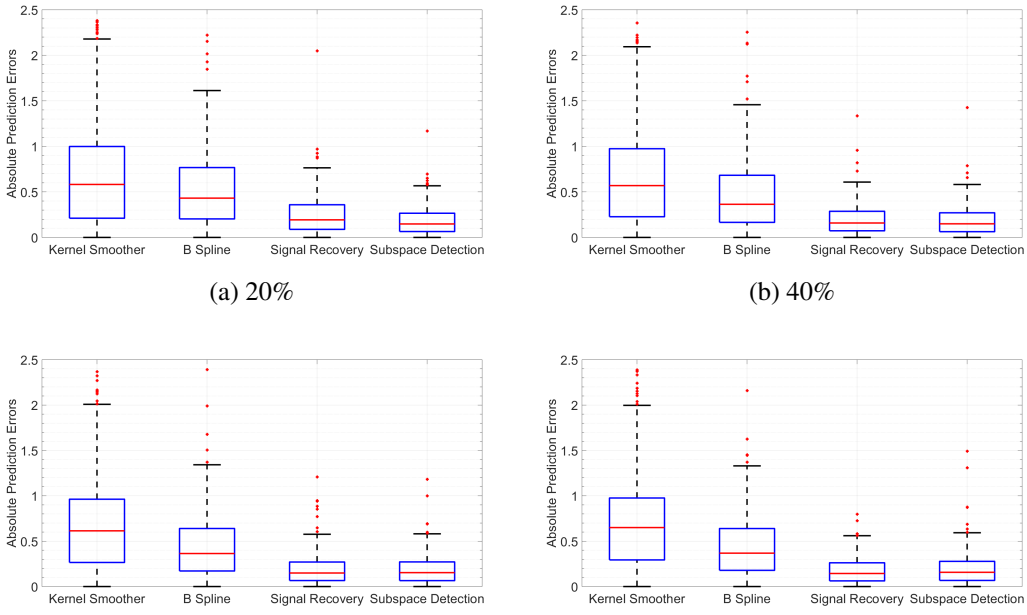


Figure 5.6: The mean prediction errors for complete signals.

Figure 5.6 illustrates the boxplots of prediction errors of our proposed methodology

and the benchmark models with complete signals. It can be seen that "Subspace detection" and "Signal recovery" perform better than both "Kernel smoother" and "B spline" in terms of prediction accuracy and precision. For example, the median (interquartile range, IQR) of the four models are 0.9 (2), 0.3 (1.1), 0.17(0.65) and 0.15 (0.55), respectively. Again, we believe this is because the proposed methodology uses more data for signal recovery than the benchmarks. Specifically, "Subspace detection" utilizes data (i), (ii), and (iii), and "Signal recovery" uses data (i) and (ii). However, "B spline" only consider data (i). Although "Kernel smoother" utilizes data (i) and (ii), it only considers local data observations neighbored to the recovered one [19].

Incomplete signals: Balance sampling



(a) 20%

(b) 40%

(c) 60%

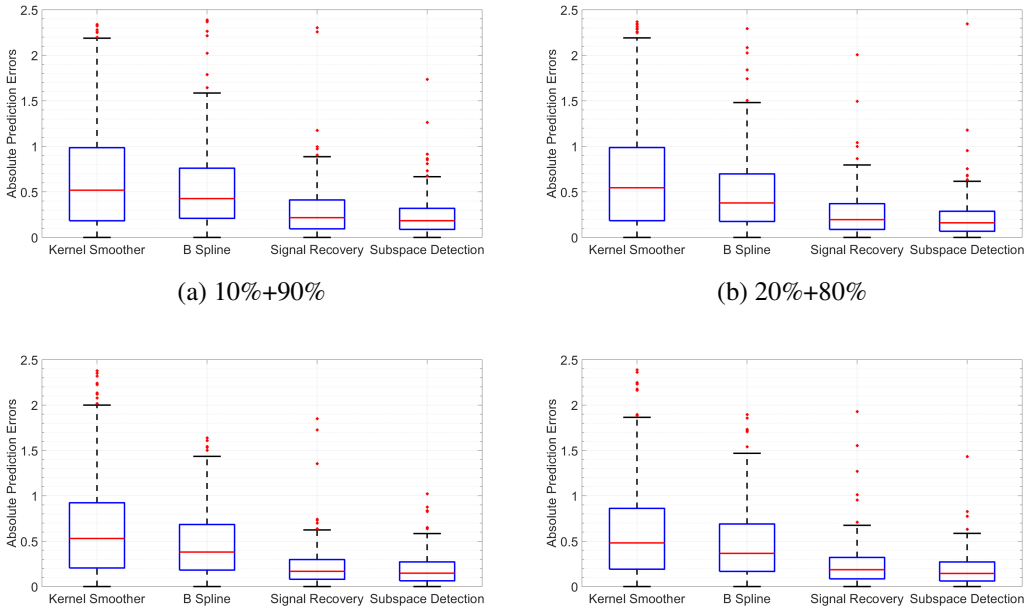
(d) 80%

Figure 5.7: The mean prediction errors with balance sampled data.

Figure 5.7 presents the prediction errors with balance sampled data. We observe that our proposed models consistently perform better than both "Kernel smoother" and "B spline" in terms of prediction accuracy and precision at all levels of data incompleteness. In addition,

we observe that "Subspace detection" outperforms "Signal recovery." For example, Figure 5.7 shows that the median (IQR) of "Subspace detection" is 0.2(0.8), while it is 0.15(0.6) for "Signal recovery." This confirms the importance of using data (iii) since "Subspace detection" utilizes data (i), (ii), and (iii), while "Signal recovery" only uses (i) and (ii).

Incomplete signals: Imbalance sampling



(a) 10%+90%

(b) 20%+80%

(c) 30%+70%

(d) 40%+60%

Figure 5.8: The mean prediction errors with imbalance sampled data.

Figure 5.8 shows the prediction errors with imbalance sampled data. Similar to the balance sampling case, we observe that our proposed models consistently perform better than the benchmarks in terms of prediction accuracy and precision at all levels of data incompleteness. This again confirms the importance of using more available data for signal recovery. In addition, we also observe that "Subspace detection" outperforms "Signal recovery." Again, we believe this affirms the importance of taking data (iii) into consideration ("Subspace detection" utilizes data (iii) while "Signal recovery" does not).

5.7 Conclusions

This chapter developed a prognostics model that fuses incomplete degradation signals from multiple sensors to predict a system's failure time in real-time. To address the incompleteness challenge, we developed two computationally efficient algorithms, both of which can extract the fused feature from the incomplete multi-stream degradation signals. These fused features are then used to predict the failure time via LLS regression. The first algorithm, the "subspace detection" method, first extracts a basis of the subspace in which the degradation signals lie by using the incomplete observations. Next, with the help of the basis, a novel feature extraction algorithm is developed to compute the singular vectors of the signal matrix. Finally, the fused features are estimated by using the singular vectors and incomplete observed signals. The second algorithm, the "signal recovery" method, first recovers the signals from each sensor via its incomplete observations. Next, the recovered signals from all the sensors are concatenated and a novel incremental SVD algorithm was developed to compute singular vectors of the concatenated signals. The incremental SVD works by adding one column of the signal matrix at a time. Finally, the fused features are estimated by using the singular vectors and incomplete signal observations.

We validated the performance of the proposed model via a simulated dataset and multi-sensor degradation data from aircraft turbofan engines. The results indicated that both of the two algorithms proposed in this chapter outperform the benchmarks in terms of prediction accuracy and precision for both complete and incomplete data. For incomplete data, we tested both balance and imbalance sampled signals. The results indicated that when the data is highly incomplete or highly imbalanced sampled, "Subspace detection" performs better than "Signal recovery." We believe this is because that "signal recovery" recovers signals one sensor at a time and does not borrow any information from the signals of other sensors who might be highly correlated with the recovered one, yet "subspace detection" simultaneously uses the information from all the sensors.

CHAPTER 6
A SUPERVISED DIMENSION REDUCTION-BASED PROGNOSTICS MODEL FOR APPLICATIONS WITH INCOMPLETE MULTI-STREAM SIGNALS AND CENSORED FAILURE TIMES

6.1 Introduction

Inexpensive sensor technology has allowed many original equipment manufacturers to install numerous sensors on their products, especially capital-intensive assets. These sensors are used to detect faults and determine the severity of an asset's degradation state through condition monitoring. Prognostics is the process of transforming raw condition monitoring data into high-fidelity degradation signals to predict the remaining useful lifetime (RUL) of an asset. However, many of these complex assets operate in harsh environments that often have a significant impact on the quality of the raw data due to errors in data acquisition, communication, read/write operations, etc. Consequently, the resulting degradation signals often contain *significant levels of missing and corrupt observations, i.e., incomplete signals (see Figure 1.5)*. In addition, in reality, historical failure times are usually *censored*. This is because equipment usually gets replaced or maintained before a failure happens, and thus no failure can be observed. Signal incompleteness and historical failure time censoring pose a significant challenge for parameter estimation of prognostic models. To address this challenge, this chapter develops a prognostic methodology that works with highly-incomplete multi-sensor degradation signals and censored historical failure times.

There are various types of prognostic models that focus on modeling multi-stream degradation signals. Examples include neural network models [22], neuro-fuzzy methods [28], and parametric models that utilize data aggregation and fusion methods [32]. However, almost all the existing models assume that the historical degradation signals are

complete, which implies that they are observed with high fidelity at frequent time steps. In addition, most of them also assume the historical failure times, also known as times-to-failure (TTFs), are *uncensored*. However, as pointed out earlier, both completeness and uncensoring assumptions are often invalid in reality. Furthermore, to fuse multi-stream degradation signals, many of the existing models employ some sort of *unsupervised dimension reduction* techniques; i.e., dimension reduction methodologies are applied to degradation signals without considering TTF information. Consequently, there is no guarantee that the extracted features are most informative for TTFs prediction.

To address the aforementioned challenges, this chapter proposes a *supervised dimension reduction (SDR)*-based prognostic methodology that is capable of modeling highly-incomplete degradation signals and censored historical TTFs. The model builds an optimization problem that combines a feature extraction term and a regression term. The feature extraction term focuses on extracting low-dimensional features using multi-stream incomplete degradation signals. It works by decomposing each system's degradation signal as a weighted combination of some unknown basis. The weights are known as the features of that system. The second term regresses the fused features against the censored TTFs via (log)-location-scale (LLS) regression. LLS regression models have been widely used in reliability engineering and survival analysis. They include a variety of TTF distributions, such as (log)normal, (log)logistic, smallest extreme value, and Weibull. The weights and basis in the first term, and the regression parameters in the second term, are estimated simultaneously from the historical dataset by solving the optimization problem mentioned earlier. Since the feature extraction process is supervised by TTFs, it is guaranteed that the extracted features are most informative for TTF prediction. To solve the optimization problem, we develop a Block Prox-Linear Coordinate Descent algorithm, which works by cyclically optimizing a block of variables at each iteration while keeping other blocks fixed. In addition, we theoretically prove the global convergence property of the algorithm.

The remainder of the chapter is organized as follows. In Section 6.2 we present the

proposed prognostics methodology. Section 6.3 introduces the optimization algorithm and discuss its convergence property. The performance of our methodology is evaluated using a numerical study in Section 6.4 and an aircraft turbofan engine degradation data in Section 7.7. Finally, Section 7.8 concludes.

6.2 Prognostic methodology

6.2.1 Model developing using incomplete signals and censored TTFs

We consider a training (historical) dataset of degradation signals for n systems. Let $\boldsymbol{s}_i \in \mathbb{R}^m$ denote the complete (i.e., no missing observations) degradation signal of system i, where m is the observation number. The m observations could be from a single sensor or concatenated from multiple sensors. Let $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)^\top \in \mathbb{R}^{n\times m}$ denote the matrix containing complete degradation signals from all the n systems. Out of the $n \times m$ entries of $\boldsymbol{S}$, we assume a subset $\Omega \subseteq \{(i,j), i = 1,\ldots,n, j = 1,\ldots,m\}$ are observed (i.e., other entries are missing). We define the projector operator $\mathcal{P}_\Omega(\cdot)$ as follows:

$$\mathcal{P}_\Omega(\boldsymbol{S})_{ij} = \begin{cases} \boldsymbol{S}_{ij}, & \text{if } (i,j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

Let the subset $O \subseteq \{(i), i = 1,\ldots,n\}$ denote the systems whose TTFs are known. In other words, we have the following:

$$\begin{cases} y_i = t_i, \text{if } i \in O \\ y_i \geq c_i, \text{if } i \notin O \end{cases} \tag{6.2}$$

where t_i and c_i are the TTF and right censored time of the ith system respectively, if its TTF follows a location-scale distribution. If its TTF comes from a log-location-scale distribution, then t_i and c_i are the logarithmic TTF and logarithmic censored time, respectively.

To fuse the multi-stream degradation signals and extract features by using their incom-

plete observations, one possible way is to optimize the following *unsupervised dimension reduction* criterion:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\mathcal{P}_{\Omega}(\boldsymbol{S}-\boldsymbol{U}\boldsymbol{V})\|_F^2 + \lambda_1(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) \tag{6.3}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ is the feature matrix, $\boldsymbol{V} \in \mathbb{R}^{p\times m}$ is the basis matrix, $\|\cdot\|_F^2$ is the Frobenius norm, p is the feature number, λ_1 is the tuning parameter. Problem (6.3) is also known as *matrix completion*, which aims to fill in the missing entries of a partially observed matrix [105]. It works by finding a minimum rank matrix that matches the observed entries. In (6.3), the first term $\min_{\boldsymbol{U},\boldsymbol{V}} \|\mathcal{P}_{\Omega}(\boldsymbol{S}-\boldsymbol{U}\boldsymbol{V})\|_F^2$ tries to find a matrix $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{V}$ best matching the observed entries of $\boldsymbol{S}$, while the second term $\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{Z}=\boldsymbol{U}\boldsymbol{V}}\{\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2\}$ is a convex relaxation of $rank(\boldsymbol{Z})$.

By optimizing (6.3), the degradation signal from the ith system (i.e., $\boldsymbol{s}_i \in \mathbb{R}^m$) is represented by its feature $\boldsymbol{u}_i \in \mathbb{R}^p$, where $\boldsymbol{u}_i$ is the ith row of the feature matrix $\boldsymbol{U}$. Usually, $p \ll m$, and thus the dimensionality of degradation signals are highly reduced. However, one limitation of criterion (6.3) is that the extracted features (i.e., $\boldsymbol{U}$) are unconnected with TTFs. In other words, there is no guarantee that the extracted features are highly correlated with TTFs. To address this challenge, we propose the following *supervised dimension reduction* methodology:

$$\begin{aligned}
&\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{y},\beta_0,\boldsymbol{\beta},\sigma} \mathcal{F}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{y},\beta_0,\boldsymbol{\beta},\sigma) \equiv \|\mathcal{P}_{\Omega}(\boldsymbol{S}-\boldsymbol{U}\boldsymbol{V})\|_F^2 + \lambda_1(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) \\
&+ w\ell(\frac{\boldsymbol{y}-\mathbf{1}_n\beta_0-\boldsymbol{U}\boldsymbol{\beta}}{\sigma}) \qquad \text{s.t.} \quad \begin{cases} y_i = t_i, & \text{if } i \in O \\ y_i \geq c_i, & \text{if } i \notin O \end{cases}
\end{aligned} \tag{6.4}$$

where $\mathbf{1}_n$ is an $n \times 1$ vector with all entries one, $\ell(\cdot)$ is the negative log-likelihood function. For example, let $\omega_i = \frac{y_i-\beta_0-\boldsymbol{u}_i\boldsymbol{\beta}}{\sigma}$, then $\ell(\frac{\boldsymbol{y}-\mathbf{1}_n\beta_0-\boldsymbol{U}\boldsymbol{\beta}}{\sigma}) = n\log\sigma - \sum_{i=1}^n \omega_i + \sum_{i=1}^n \exp(\omega_i)$ for an SEV/Weibull distribution, $\ell(\frac{\boldsymbol{y}-\mathbf{1}_n\beta_0-\boldsymbol{U}\boldsymbol{\beta}}{\sigma}) = n\log\sigma - \sum_{i=1}^n \omega_i + 2\sum_{i=1}^n \log(1+$

$\exp(\omega_i))$ for a logistics/loglogistics distribution, $\ell(\frac{\boldsymbol{y}-\mathbf{1}_n\beta_0-\boldsymbol{u}_i\beta}{\sigma}) = \frac{n}{2}\log 2\pi + n\log\sigma + \frac{1}{2}\sum_{i=1}^{n}\omega_i^2$ for a normal/lognormal distribution. w is a weight controlling the balance between the unsupervised term and supervised term.

One limitation of optimization problem (6.4) is that it is not block multi-convex. An optimization problem is block multi-convex when its feasible set and objective function are generally non-convex but convex in each block of variables. The block multi-convex property can significantly simplify a non-convex optimization problem such as the one in criterion (6.4) [106]. Therefore, we apply the following re-parameterization to transform (6.4) to a block multi-convex problem: $\tilde{\sigma} = 1/\sigma, \tilde{\beta}_0 = \beta_0/\sigma, \tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}/\sigma$. As a result, criterion (6.4) is re-expressed as follows:

$$
\begin{aligned}
&\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{y},\tilde{\beta}_0,\tilde{\boldsymbol{\beta}},\tilde{\sigma}} \tilde{\mathcal{F}}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{y},\tilde{\beta}_0,\tilde{\boldsymbol{\beta}},\tilde{\sigma}) \\
&\equiv \|\mathcal{P}_\Omega(\boldsymbol{S}-\boldsymbol{U}\boldsymbol{V})\|_F^2 + \lambda_1(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) + w\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) + \lambda_2\|\tilde{\boldsymbol{\beta}}\|_1, \\
&\text{s.t.} \quad \begin{cases} y_i = t_i, & \text{if } i \in O \\ y_i \geq c_i, & \text{if } i \notin O \end{cases}
\end{aligned}
\tag{6.5}
$$

Let $\tilde{\omega}_i = \tilde{\sigma}y_i - \tilde{\beta}_0 - \boldsymbol{u}_i\tilde{\boldsymbol{\beta}}$, then $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = -n\log\tilde{\sigma} - \sum_{i=1}^{n}\tilde{\omega}_i + \sum_{i=1}^{n}\exp(\tilde{\omega}_i)$ for an SEV/Weibull distribution, $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = -n\log\tilde{\sigma} - \sum_{i=1}^{n}\tilde{\omega}_i + 2\sum_{i=1}^{n}\log(1+\exp(\tilde{\omega}_i))$ for a logistics/loglogistics distribution, $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = \frac{n}{2}\log 2\pi - n\log\tilde{\sigma} + \frac{1}{2}\sum_{i=1}^{n}\tilde{\omega}_i^2$ for a normal/lognormal distribution. Notice that in (6.5), a penalty term $\|\tilde{\boldsymbol{\beta}}\|_1$ is added to inspire sparsity and avoid overfitting. The algorithm to solve problem (6.5) will be discussed in Section 6.3.

6.2.2 Real-time RUL prediction

Our goal is to predict and update the RUL of partially degraded systems that are still operating in the field. To achieve this goal, we first optimize problem (6.5) using the training

dataset. As a result, the SDR basis $\hat{\boldsymbol{V}}$, regression coefficient $\hat{\tilde{\beta}}_0$ and $\hat{\tilde{\boldsymbol{\beta}}}$, and the scale parameter $\hat{\tilde{\sigma}}$ are estimated. Next, the real-time degradation signals observed from the field system are used to extract features. Finally, the extracted features are input to the LLS regression model to predict TTF. RUL is obtained by subtracting the current observation time.

Denote the real-time degradation signal by $\boldsymbol{s}_i^{\text{new}}$ and only a subset $\Omega^{\text{new}} \subseteq \{(j), j = 1, \ldots, m\}$ are observed. Based on the promise that $\boldsymbol{s}_i^{\text{new}}$ can be expressed as linear combination of the SDR basis $\hat{\boldsymbol{V}}$, the feature of the real-time observed degradation signals can be computed by optimizing the following criterion:

$$\boldsymbol{u}^{\text{new}} = \min_{\boldsymbol{u}^{\text{new}}} \|\mathcal{P}_{\Omega^{\text{new}}}(\hat{\boldsymbol{V}})^\top \boldsymbol{u}^{\text{new}} - \mathcal{P}_{\Omega^{\text{new}}}(\boldsymbol{s}_i^{\text{new}})\|_2^2 \tag{6.6}$$

where $\boldsymbol{u}^{\text{new}} \in \mathbb{R}^p$ is the extracted feature, $\mathcal{P}_{\Omega^{\text{new}}}(\hat{\boldsymbol{V}}) \in \mathbb{R}^{p \times |\Omega^{\text{new}}|}$ consists the $|\Omega^{\text{new}}|$ rows of matrix $\hat{\boldsymbol{V}}$ indexed by the set Ω^{new} and $\mathcal{P}_{\Omega^{\text{new}}}(\boldsymbol{s}_i^{\text{new}}) \in \mathbb{R}^{|\Omega^{\text{new}}|}$ is the observed subset of the real-time signals.

Finally, the predicted TTF can be computed by using $\boldsymbol{u}^{\text{new}}$ and the estimated LLS regression model. For example, for Weibull and lognormal regression, the predicted TTF are $\exp(\hat{\beta}_0 + \boldsymbol{u}^{\text{new}}\hat{\boldsymbol{\beta}})\Gamma(1 + \hat{\sigma})$ and $\exp(\hat{\beta}_0 + \boldsymbol{u}^{\text{new}}\hat{\boldsymbol{\beta}} + \hat{\sigma}^2/2)$, respectively, where $\Gamma(\cdot)$ is the Gamma function, $\hat{\beta}_0 = \hat{\tilde{\beta}}_0/\hat{\tilde{\sigma}}$, $\hat{\boldsymbol{\beta}} = \hat{\tilde{\boldsymbol{\beta}}}/\hat{\tilde{\sigma}}$ and $\hat{\sigma} = 1/\hat{\tilde{\sigma}}$.

6.3 Optimization algorithm

In this section, we propose a Block Prox-Linear Coordinate Descent (BPLCD) algorithm to solve optimization criterion (6.5) and prove that the proposed BPLCD algorithm has a global convergence property.

6.3.1 Block prox-linear coordinate descent

The BPLCD algorithm works by cyclically updating a block of variables at each iteration by minimizing a prox-linear surrogate function. Specifically, at the kth iteration, $\boldsymbol{U}$ is

updated by solving the following optimization problem while keeping other blocks (i.e., $\boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}$) fixed:

$$\boldsymbol{U}^k = \min_{\boldsymbol{U}} \langle \nabla f^k(\hat{\boldsymbol{U}}^k), \boldsymbol{U} - \hat{\boldsymbol{U}}^k \rangle + \frac{L_{\boldsymbol{U}}^k}{2} \|\boldsymbol{U} - \hat{\boldsymbol{U}}^k\|_2^2 + \lambda_1 \|\boldsymbol{U}\|_F^2 \tag{6.7}$$

where function $f(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}) = \|\mathcal{P}_\Omega(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{V})\|_F^2 + w\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}), \nabla f^k(\hat{\boldsymbol{U}}^k)$ is the block-partial gradient of f at $\hat{\boldsymbol{U}}^k$, $\hat{\boldsymbol{U}}^k = \boldsymbol{U}^{k-1} + \omega_{\boldsymbol{U}}^k(\boldsymbol{U}^{k-1} - \boldsymbol{U}^{\text{prev}})$ denotes an extrapolated point, $\omega_{\boldsymbol{U}}^k \geq 0$ is the extrapolation weight and $\boldsymbol{U}^{\text{prev}}$ is the value of $\boldsymbol{U}$ before it was updated to $\boldsymbol{U}^{k-1}$. The extrapolation weight $\omega_{\boldsymbol{U}}^k$ can be simply set as 0, while an appropriate $\omega_{\boldsymbol{U}}^k > 0$ can significantly accelerate the convergence of our algorithm. L^k is a constant controls the step size. Usually, $L_{\boldsymbol{U}}^k$ can be set as $\alpha\tilde{L}_{\boldsymbol{U}}^k$ with any $\alpha > 1$, where $\tilde{L}_{\boldsymbol{U}}^k$ is the Lipschitz constant of $\nabla f^k(\boldsymbol{U})$. More details regarding the selecting of ω_k and $L_{\boldsymbol{U}}^k$ can be found in [106]. Similarly, the remaining blocks of variable are updated as follows:

$$\boldsymbol{V}^k = \min_{\boldsymbol{V}} \langle \nabla f^k(\hat{\boldsymbol{V}}^k), \boldsymbol{V} - \hat{\boldsymbol{V}}^k \rangle + \frac{L_{\boldsymbol{V}}^k}{2} \|\boldsymbol{V} - \hat{\boldsymbol{V}}^k\|_2^2 + \lambda_1 \|\boldsymbol{V}\|_F^2 \tag{6.8}$$

$$\tilde{\boldsymbol{\beta}}^k = \min_{\tilde{\boldsymbol{\beta}}} \langle \nabla f^k(\hat{\boldsymbol{\beta}}^k), \tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^k \rangle + \frac{L_{\tilde{\boldsymbol{\beta}}}^k}{2} \|\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^k\|_2^2 + \lambda_2 \|\tilde{\boldsymbol{\beta}}\|_1 \tag{6.9}$$

$$\tilde{\beta}_0^k = \min_{\tilde{\beta}_0} \langle \nabla f^k(\hat{\beta}_0^k), \tilde{\beta}_0 - \hat{\beta}_0^k \rangle + \frac{L_{\tilde{\beta}_0}^k}{2} \|\tilde{\beta}_0 - \hat{\beta}_0^k\|_2^2 \tag{6.10}$$

$$\tilde{\sigma}^k = \min_{\tilde{\sigma}} \langle \nabla f^k(\hat{\sigma}^k), \tilde{\sigma} - \hat{\sigma}^k \rangle + \frac{L_{\tilde{\sigma}}^k}{2} \|\tilde{\sigma} - \hat{\sigma}^k\|_2^2 \tag{6.11}$$

$$\tilde{\boldsymbol{y}}^k = \min_{\tilde{\boldsymbol{y}}} \langle \nabla f^k(\hat{\boldsymbol{y}}^k), \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}}^k \rangle + \frac{L_{\tilde{\boldsymbol{y}}}^k}{2} \|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}}^k\|_2^2,$$
$$\text{subject to} \quad \begin{cases} {y_i}^k = t_i, & \text{if } i \in O \\ {y_i}^k \geq c_i, & \text{if } i \notin O \end{cases} \tag{6.12}$$

where $\hat{\boldsymbol{V}}^k = \boldsymbol{V}^{k-1} + \omega_{\boldsymbol{V}}^k(\boldsymbol{V}^{k-1} - \boldsymbol{V}^{\text{prev}})$, $\hat{\boldsymbol{\beta}}^k = \tilde{\boldsymbol{\beta}}^{k-1} + \omega_{\tilde{\boldsymbol{\beta}}}^k(\tilde{\boldsymbol{\beta}}^{k-1} - \tilde{\boldsymbol{\beta}}^{\text{prev}})$, $\hat{\beta}_0^k = \tilde{\beta}_0^{k-1} + \omega_{\tilde{\beta}_0}^k(\tilde{\beta}_0^{k-1} - \tilde{\beta}_0^{\text{prev}})$, $\hat{\sigma}^k = \tilde{\sigma}^{k-1} + \omega_{\tilde{\sigma}}^k(\tilde{\sigma}^{k-1} - \tilde{\sigma}^{\text{prev}})$ and $\hat{\boldsymbol{y}}^k = \boldsymbol{y}^{k-1} + \omega_{\boldsymbol{y}}^k(\boldsymbol{y}^{k-1} - \boldsymbol{y}^{\text{prev}})$. We summarize the optimization algorithm in Algorithm 3. Usually, the stopping criterion can be set as $\tilde{\mathcal{F}}(\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{y}^k, \tilde{\beta}_0^k, \tilde{\boldsymbol{\beta}}^k, \tilde{\sigma}^k) - \tilde{\mathcal{F}}(\boldsymbol{U}^{k+1}, \boldsymbol{V}^{k+1}, \boldsymbol{y}^{k+1}, \tilde{\beta}_0^{k+1}, \tilde{\boldsymbol{\beta}}^{k+1}, \tilde{\sigma}^{k+1}) < \epsilon$, where ϵ is a small number, say $1e-3$.

Algorithm 3: Block Prox-Linear Coordinate Descent (BPLCD) algorithm for solving problem (6.5)

Initialization: Choose two initial points
$\boldsymbol{U}^{-1} = \boldsymbol{U}^0, \boldsymbol{V}^{-1} = \boldsymbol{V}^0, \boldsymbol{\beta}^{-1} = \boldsymbol{\beta}^0, \tilde{\sigma}^{-1} = \tilde{\sigma}^0, {y_i}^{-1} = {y_i}^0 \geq c_i$ for $i \notin O$
for $k = 1, 2, \ldots$
  $\boldsymbol{U}^k \leftarrow$ (6.7)
  $\boldsymbol{V}^k \leftarrow$ (6.8)
  $\tilde{\boldsymbol{\beta}}^k \leftarrow$ (6.9)
  $\tilde{\beta}_0^k \leftarrow$ (6.10)
  $\tilde{\sigma}^k \leftarrow$ (6.11)
  $\boldsymbol{y}^k \leftarrow$ (6.12)
  if stopping criterion is satisfied
    return $\{\boldsymbol{U}^k, \boldsymbol{V}^k, \tilde{\beta}_0^k, \tilde{\boldsymbol{\beta}}^k, \tilde{\sigma}^k, \boldsymbol{y}^k\}$
  end if
end for

6.3.2 Convergence property

In this section, we discuss the convergence property of Algorithm 3. Denote the variables in optimization problem (6.5) by $\boldsymbol{\theta} = \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}\}$ and the sequence generated by Algorithm 3 by $\{\boldsymbol{\theta}^k\}_{k\geq 1}$. We first prove that $\{\boldsymbol{\theta}^k\}_{k\geq 1}$ has a finite limit point. To do this,

we give the following lemma.

Lemma 6.3.1 *For optimization problem (6.5), if $\tilde{\mathcal{F}}(\boldsymbol{\theta}^k) \leq \tilde{\mathcal{F}}(\boldsymbol{\theta}^0)$ where $\boldsymbol{\theta}^0$ is any feasible solution, then $\boldsymbol{\theta}^k$ is bounded, i.e., there exist constants $M_{\boldsymbol{U}}$, $M_{\boldsymbol{V}}$, $M_{\tilde{\beta}_0}$, $M_{\tilde{\boldsymbol{\beta}}}$, M_{y_i}, $M_{\tilde{\sigma}}^{\min}$, $M_{\tilde{\sigma}}^{\max}$ such that $\|\boldsymbol{U}^k\|_F^2 \leq M_{\boldsymbol{U}}$, $\|\boldsymbol{V}^k\|_F^2 \leq M_{\boldsymbol{V}}$, $\|\tilde{\beta}_0^k\|_1 \leq M_{\tilde{\beta}_0}$, $\|\tilde{\boldsymbol{\beta}}^k\|_1 \leq M_{\tilde{\boldsymbol{\beta}}}$, $c_i < y_i^k < M_{y_i}, \forall i \notin O$ and $0 < M_{\tilde{\sigma}}^{\min} \leq \tilde{\sigma}^k \leq M_{\tilde{\sigma}}^{\max}$*

The proof of Lemma 6.3.1 can be found in the appendix. Lemma 6.3.1 implies that $\{\boldsymbol{\theta}^k\}_{k\geq 1}$ has a bounded subsequence. Therefore, we have the following proposition.

Proposition 3 *The sequence $\{\boldsymbol{\theta}^k\}_{k\geq 1}$ generated by Algorithm 3 has a finite limit point $\boldsymbol{\theta}^*$.*

Next, we prove that the objective function $\tilde{\mathcal{F}}$ in optimization criterion (6.5) satisfies the Kurdyka-Łojasiewicz (KL) property around $\boldsymbol{\theta}^*$. The KL property has been widely used recently in proving the convergence of non-convex optimization problems [106]. The definition of KL property is given below.

Definition 6.3.1 *(Kurdyka-Łojasiewicz property [106]) A function $\psi(\boldsymbol{x})$ satisfies the KL property at point $\boldsymbol{x}^* \in dom(\partial\psi)$ if there exist $\eta > 0$, a neighborhood $\mathcal{B}_\rho(\boldsymbol{x}^*) \triangleq \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^*\| < \rho\}$, and a concave function $\phi(a) = b \cdot a^{1-\gamma}$ for some $b > 0$ and $\gamma \in [0, 1)$ such that the KL inequality holds:*

$$\phi'(|\psi(\boldsymbol{x}) - \psi(\boldsymbol{x}^*)|)dist(\mathbf{0}, \partial\psi(\boldsymbol{x})) \geq 1, \text{ for any } \boldsymbol{x} \in$$
$$\mathcal{B}_\rho(\boldsymbol{x}^*) \cap dom(\partial\psi) \text{ and } \psi(\boldsymbol{x}^*) < \psi(\boldsymbol{x}) < \psi(\boldsymbol{x}) + \eta,$$

where $dom(\partial\psi) = \{\boldsymbol{x} : \partial\psi(\boldsymbol{x}) \neq \emptyset\}$ and $dist(\mathbf{0}, \partial\psi(\boldsymbol{x})) = \min\{\|\boldsymbol{z}\| : \boldsymbol{z} \in \partial\psi(\boldsymbol{x})\}$.

Lemma 6.3.2 *The objective function $\tilde{\mathcal{F}}$ in optimization criterion (6.5) satisfies the Kurdyka-Łojasiewicz property around $\boldsymbol{\theta}^*$.*

The proof of Lemma 6.3.2 is given in the appendix. Finally, we prove that the block-partial gradient of the function $f(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}) = \|\mathcal{P}_\Omega(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{V})\|_F^2 + w\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}})$,

where ℓ is the negative log-likelihood function from an LLS distribution, are Lipschitz continuous in a bounded set.

Lemma 6.3.3 *For LLS distributions, if $\boldsymbol{\theta}^k$ is bounded, then the block-partial gradient $\nabla f^k(\boldsymbol{U}^k)$, $\nabla f^k(\boldsymbol{V}^k)$, $\nabla f^k(\tilde{\beta}_0^k)$, $\nabla f^k(\tilde{\boldsymbol{\beta}}^k)$, $\nabla f^k(\tilde{\sigma}^k)$, $\nabla f^k(\boldsymbol{y}^k)$ are Lipschitz continuous in which bounded set.*

The proof of Lemma 6.3.3 is also shown in the appendix. From Lemma 6.3.1, we know that $\boldsymbol{\theta}^k$ is bounded. Therefore, $\nabla f^k(\boldsymbol{U}^k)$, $\nabla f^k(\boldsymbol{V}^k)$, $\nabla f^k(\tilde{\beta}_0^k)$, $\nabla f^k(\tilde{\boldsymbol{\beta}}^k)$, $\nabla f^k(\tilde{\sigma}^k)$, $\nabla f^k(\boldsymbol{y}^k)$ are Lipschitz continuous.

Given Proposition 3, Lemma 6.3.2 and Lemma 6.3.3, we are ready to give the convergence property of Algorithm 3 in the following Theorem.

Theorem 6.3.1 *(Global convergence) The sequence $\{\boldsymbol{\theta}^k\}_{k\geq 1}$ generated by Algorithm 3 converges to $\boldsymbol{\theta}^* = (\boldsymbol{U}^*, \boldsymbol{V}^*, \boldsymbol{y}^*, \tilde{\beta}_0^*, \tilde{\boldsymbol{\beta}}^*, \tilde{\sigma}^*)$, which is a critical point of optimization problem (6.5).*

The proof of Theorem 6.3.1 is shown in the appendix.

6.4 Simulation study

In this section, we conduct a simulation study to validate the proposed prognostic methodology. We compare the performance of our methodology, designated "SDR," with two baseline methods. The first baseline method is designated "Matrix completion." It works by first applying a matrix completion technique [107] to recover the missing observations of degradation signals. Next, an unsupervised dimension reduction technique–multivariate functional principal component analysis–is applied to the recovered degradation signals to extract fused features. Finally, the fused features are regressed against TTFs using LLS regression. The second benchmarking model, designated "B Spline," is similar to "Matrix completion" except that the degradation signals are recovered using penalized B Spline.

The tuning parameter of penalized B spline is selected by utilizing generalized cross validation (GCV).

We evaluate the prediction errors of our model and the benchmarking models at two levels of data incompleteness: 20% and 80%, where 20% means that 20% of the observations are randomly missing. At each data incompleteness level, we consider two levels of TTF censoring: 20% and 80%. Taking 20% as an example, the TTFs are censored as follows: First, 20% of the systems are randomly selected from the training dataset. Next, the smallest TTF of the selected systems is set as the censored TTFs of all the selected systems. The prediction errors are calculated from Equation (2.25). We report the prediction errors at the following life percentiles: $10th$, $20th$, ..., $90th$, where for example the $10th$ percentile represents the average prediction errors evaluated at life percentiles in the interval of $(5\%, 15\%]$, $20th$ for the interval of $(15\%, 25\%]$, etc.

6.4.1 Generating degradation signals

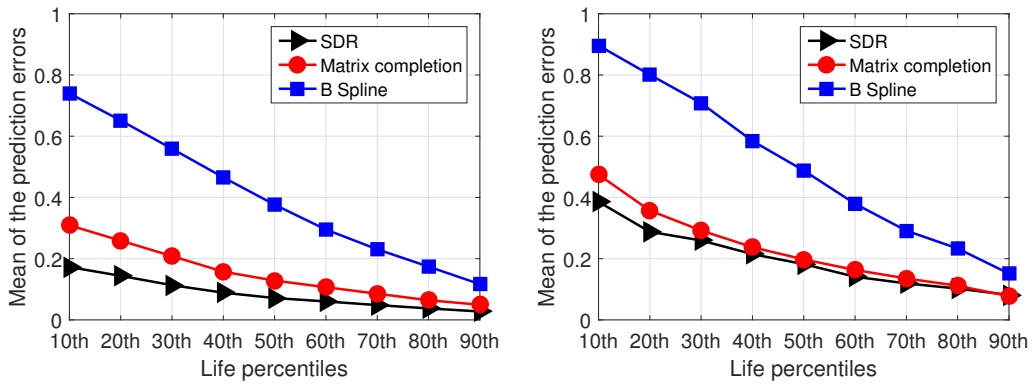
In this simulation, we consider 500 identical systems, each of which is assumed to be monitored by 5 sensors. For each system $i; i = 1, \ldots, 500$, we begin by simulating its underlying degradation path using the following functional form: $\boldsymbol{s}_i^u(t) = \xi_{i,1}\phi_1(t) + \xi_{i,2}\phi_2(t)$, where $\phi_1(t) = t/\sin(t)$ and $\phi_2(t) = t^2/\sin(t)$ are two basis; $\{\xi_{i,1}\}_{i=1}^{500}$ are randomly generated from $\mathcal{N}(0.1, 0.01^2)$ and then sorted in ascending order. Similarly, $\{\xi_{i,2}\}_{i=1}^{500}$ are randomly generated from $\mathcal{N}(0.2, 0.03^2)$ and then also sorted in ascending order. Next, we generate the TTFs of these simulated systems by using the following equation: $\log(t_i) = \beta_1\xi_{i,1} + \beta_2\xi_{i,2} + \epsilon_i$, where $\beta_1 = 1, \beta_2 = 2$ and $\epsilon_i \sim \mathcal{N}(0, 0.0001^2)$. It can be seen that the generated TTFs follows a lognormal distribution. Finally, we generate the degradation signals of the 5 sensors as follows: $\boldsymbol{s}_i(t) = \boldsymbol{s}_i^u(t) + \varepsilon_i(t)$, where $\varepsilon_i(t) \sim \mathcal{N}(0, 0.02^2)$ is IID noise.

The whole simulation procedure is replicated 100 times. For each replication, 400 systems are randomly selected for training and the remaining 100 systems are used for

testing.

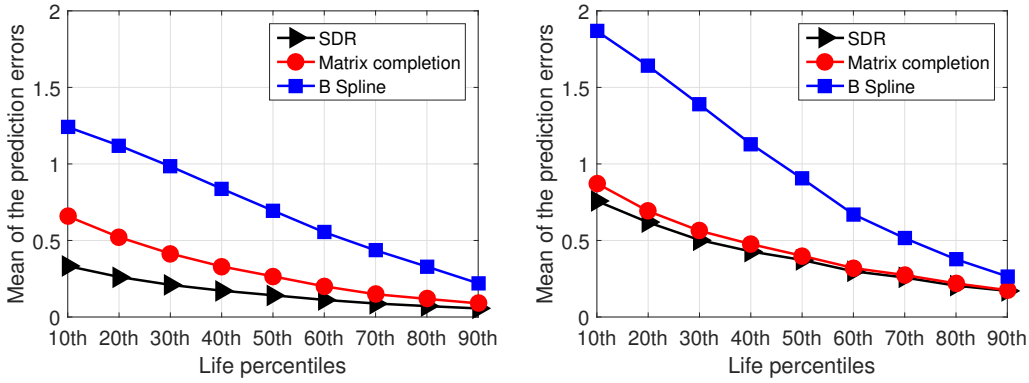
6.4.2 Results and analysis

The mean of absolute prediction errors when 20% and 80% degradation observations missing are reported in Figures 6.1 and 6.2, respectively.



(a) 20% TTF censored (b) 80% TTF censored

Figure 6.1: Prediction errors when 20% observations of the degradation signals are missing.



(a) 20% TTF censored (b) 80% TTF censored

Figure 6.2: Prediction errors when 80% observations of the degradation signals are missing.

Figures 6.1 and 6.2 indicated that all three methodologies achieved better prediction accuracy at a lower level of signal incompleteness. For example, Plot (a) in Figure 6.1 shows that, at the $50th$ life percentiles, the prediction errors of "SDR," "Matrix completion," and "B Spline" are $0.4, 0.15, 0.1$, respectively. However, they are $0.7, 0.25, 0.15$ respectively in

Plot (a) of Figure 6.2. Similarly, all the three models had better prediction accuracy at a lower level of TTF censoring. This is reasonable since less missing observations or censored TTFs means more information is available to use, and thus lower prediction errors are achieved.

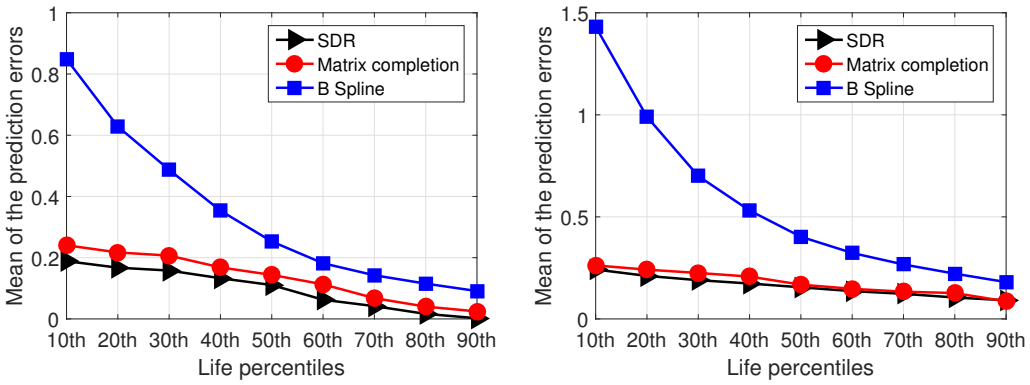
Figures 6.1 and 6.2 indicated that at all levels of data incompleteness and TTF censoring, "B Spline" did not perform well as "SDR" and "Matrix completion." We believe this is because "B Spline" recovers the missed observations of each degradation signal by fitting that signal individually, which means no information from other degradation signals is used. However, "SDR" and "Matrix completion" recover the missed observations of all the signals simultaneously, and thus more information is used to imputed each missed observation.

Figures 6.1 and 6.2 also illustrated that our proposed "SDR" outperforms "Matrix completion," which confirmed the importance of using TTFs to supervise the dimension reduction of degradation signals. Plot 2 (a) illustrated that the prediction errors of "SDR" are much smaller than that of "Matrix completion" when 20% TTFs are censored, while Plot 2 (b) showed the prediction errors between the two models are very close when 80% TTFs are censored. This is reasonable because Plot 2 (a) used more TTF information to supervise the dimension reduction process than Plot 2 (b), so "SDR" achieved better prediction accuracy than Plot 2 (b). A similar phenomenon can also be observed in Figure 6.2.

6.5 Case study

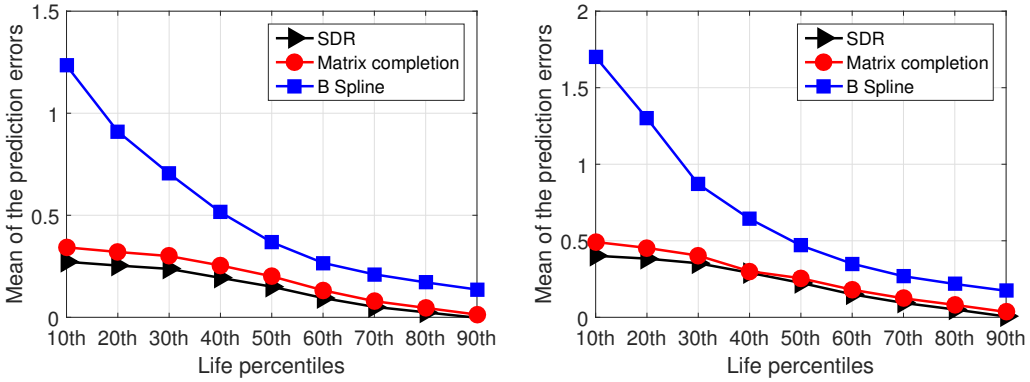
In this section, we use multi-sensor degradation data from aircraft turbofan engines provided by NASA [11] to evaluate the performance of our model. The dataset is comprised of the following; (i) degradation signals from 100 training engines that were run to failure, (ii) degradation signals from an additional 100 test engines whose operation was prematurely terminated at random time points prior to their failure time, and (iii) the real TTFs of the 100 test engines. Each engine was monitored using 21 sensors.

We evaluate the performance of our model and the two benchmarks following a similar manner to the simulation study. The prediction errors are reported in Figures 6.3 and 6.4.



(a) 20% TTF censored (b) 80% TTF censored

Figure 6.3: Prediction errors when 20% observations of the degradation signals are missing.



(a) 20% TTF censored (b) 80% TTF censored

Figure 6.4: Prediction errors when 80% observations of the degradation signals are missing.

Figures 6.3 and 6.4 indicated that "SDR" and "Matrix completion" worked better than "B Spline" at all levels of data incompleteness and TTF censoring. We again believe this is because "B Spline" fits each degradation signals in an individual manner and uses no information from other degradation signals, while "SDR" and "Matrix completion" use more information to impute each missed observation by recovering it using all available signal observations in the historical dataset.

Figures 6.3 and 6.4 also indicated that the prediction errors of our proposed methodology are smaller than that of "Matrix completion," which again confirmed the importance of using TTFs to supervise the dimension reduction of degradation signals. Moreover, Figures 6.3 (a) illustrated that the difference of prediction errors between our methodology and "Matrix completion" is obvious. However, the difference between them in Figures 6.3 (b) is negligible. A similar phenomenon can also be observed in Figure 6.4. This is reasonable since the prediction errors in Figures 6.3 (a) were evaluated when 20% TTFs were censored, while in Figures 6.3 (b) they were evaluated when 80% TTFs were censored. Since Figures 6.3 (a) used more TTF information to supervise the dimension reduction process than Figures 6.3 (b), "SDR" achieved obviously better prediction accuracy than "Matrix completion." In contrast, since Figures 6.3 (b) used less TTF information, and thus the prediction accuracy between "SDR" and "Matrix completion" are close.

6.6 Conclusions

This chapter developed a prognostic methodology for multi-sensor applications with *highly-incomplete* degradation signals and *censored* historical failure times. The methodology builds an optimization problem combining a feature extraction term and a regression term. The feature extraction term extracts low-dimensional features of multi-stream degradation signals using their incomplete observations, and the regression term regresses the features against the censored TTFs. By simultaneously optimizing the two terms, the TTFs are used to supervise the feature extraction process, and thus the extracted features are guaranteed to be most informative for TTF prediction. To solve the optimization problem, we developed a Block Prox-Linear Coordinate Descent algorithm and theoretically proved its global convergence property.

A simulated dataset and a multi-stream degradation data from aircraft turbofan engines were used to evaluate the performance of our proposed methodology. The results indicated that our proposed methodology achieved high prediction accuracy even if the degradation

signals are highly incomplete and the historical failure times present a significant level of censoring. In addition, the results also illustrated that our model consistently outperformed the unsupervised dimension reduction-based benchmarks in terms of prediction errors, at different levels of data incompleteness and failure time censoring, which confirmed the importance of using failure times to supervise the feature extraction (or dimension reduction) process in prognostic modeling.

CHAPTER 7

RESIDUAL USEFUL LIFETIME PREDICTION USING A DEGRADATION IMAGE STREAM VIA PENALIZED TENSOR REGRESSION

7.1 Introduction

Imaging is one of the fastest growing technologies for condition monitoring and industrial asset management. Relative to most sensing techniques, industrial imaging devices are easier to use because they are generally noncontact and do not require permanent installation or fixturing. Image data also contains rich information about the object being monitored. Some examples of industrial imaging technologies include infrared images used to measure temperature distributions of equipment and components [108], charge-coupled device (CCD) images which capture surface quality information (e.g., cracks) of products [109], and others. Image data has been extensively used for process monitoring and diagnostics. For instance, infrared images have been successfully used for civil structures monitoring [110], machinery inspection [111], fatigue damage evaluation [112] and electronic printed circuit board (PCB) monitoring [113]. In steel industry, CCD cameras have been utilized for product surface inspection [109], while video cameras have been used to monitor the shape and color of furnace flames to control quality of steel tubes [114]. This chapter expands the utilization of image data by proposing an image-based prognostic modeling framework that uses degradation-based image streams to predict remaining lifetime.

Numerous prognostic methodologies have been developed in the literature. Examples of some modeling approaches include random coefficients models [1, 2], models that utilize the Brownian motion process [4, 5] and gamma process [7, 8], and models based on functional data analysis [19, 93]. These approaches are well-suited for time-series signals, but it is not clear how they can be extended to model image streams. One of the key challenges

in modeling image data revolves around the analytical and computational complexities associated with characterizing high dimensional data. High dimensionality arises from the fact that a single image stream consists of a large sequence of images (observed across the life cycle of an equipment) coupled with the large numbers of pixels embedded in each image. Additional challenges are related to the complex *spatial-temporal structures* inherent in the data streams. Pixels are spatially correlated within a single image and temporally correlated across sequential images. In recent work [115], degradation image streams were modeled as a spatio-temporal process. Although spatio-temporal models have been widely used to model data with complex spatial and temporal correlation structures [116], they are not necessarily accurate for long-term predictions necessary to our type of prognostic application. Most importantly, a key limitation of spatio-temporal models is that they require a pre-set failure threshold, which is usually hard to define for degradation image streams.

This chapter proposes a tensor-based regression framework that utilizes degradation image streams to predict remaining useful life, and provide advance warning of impending failures of industrial assets. Specifically, we build a LLS tensor regression model in which the TTF is treated as the response and degradation image streams as covariates. To model the *spatio-temporal structure* of degradation image streams, the regression model treats each image stream as a *tensor*. A tensor is defined as a *multi-dimensional array*–a one-order tensor is a vector, a two-order tensor is a matrix, and objects of order three or higher are called high-order tensors. More details about tensor theory and applications can be found in a survey paper by [117]. As illustrated in Figure 7.1, a degradation image stream constitutes a three-order tensor in which the first two dimensions capture the spatial structure of a single image whereas the third dimension is used to model the temporal structure of the image stream. One of the key benefits of modeling a degradation image stream as a tensor is that tensors maintain the *spatio-temporal structure* within and between images which allows for a relatively accurate RUL prediction model. In this chapter, *degradation image stream(s)* and *degradation tensor(s)* are used exchangeably hereafter.

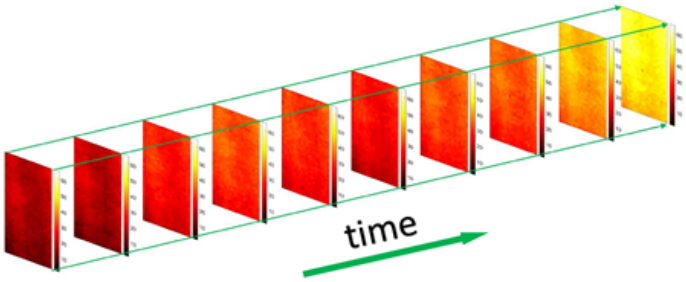


Figure 7.1: An illustration of a degradation image stream (3-order tensor).

The high dimensionality of degradation image streams poses significant computational challenges, especially ones related to parameter estimation. For example, a tensor-regression model for a degradation image stream consisting of 50 images each with 20×20 pixels generates a three-order tensor-coefficient consisting of 20,000 elements that need to be estimated. In an effort to improve model computations, we develop two estimation methods that integrate dimensionality reduction and tensor decomposition. Dimensionality reduction is used as the first step for both estimation methods as it helps reduce the number of parameters. Degradation tensors are projected to a low-dimensional tensor subspace that preserves most of their information. This is achieved using a multilinear dimension reduction technique, such as multilinear principal component analysis (MPCA) [21]. We utilize the fact that essential information embedded in high-dimensional tensors can be captured in a low-dimensional tensor subspace. Next, the tensor-coefficients corresponding to the projected degradation tensors are decomposed using two popular tensor decomposition approaches namely, CANDECOMP/PARAFAC (CP) [118] and Tucker [119]. The CP approach decomposes a high-dimensional coefficient tensor as a product of several low-rank basis matrices. In contrast, the Tucker approach expresses the tensor-coefficient as a product of a low-dimensional core tensor and several factor matrices. Therefore, instead of estimating the tensor-coefficient, we only estimate its corresponding core tensors and factor/basis matrices, which significantly reduces the computational complexity and the required sample size. Block relaxation algorithms are also developed for model estimation with guaranteed global convergence to a stationary point.

The remainder of the chapter is organized as follows. Section 7.2 provides an overview of the basic notations and definitions in multilinear algebra. Section 7.3 presents the degradation and prognostic modeling framework. Sections 7.3.1 and 7.3.2 discuss the estimation algorithm based on CP decomposition and Tucker decomposition, respectively. In Section 7.4, we discuss the RUL prediction and realtime updating. The effectiveness of our model is validated using simulated data in Sections 7.5 and 7.6 along with real degradation image streams from a rotating machinery in Section 7.7. Finally, Section 7.8 is devoted to concluding remarks.

7.2 Preliminaries

This section presents some basic notations, definitions and operators in multilinear algebra and tensor analysis that are used throughout the chapter. Scalars are denoted by lowercase letters, e.g., b, vectors are denoted by lowercase boldface letters, e.g., $\boldsymbol{b}$, matrices are denoted by uppercase boldface letters, e.g., $\boldsymbol{B}$, and tensors are denoted by calligraphic letters, e.g., $\mathcal{B}$. The *order* of a tensor is the number of modes, also known as *way*. For example, the order of vectors and matrices are 1 and 2, respectively. A D-order tensor is denoted by $\mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$, where I_d for $d = 1, \ldots, D$ represents the dimension of the d-mode of $\mathcal{B}$. The $(i_1, i_2, \ldots, i_D)$-th component of $\mathcal{B}$ is denoted by $b_{i_1,i_2,\ldots,i_D}$. A *fiber* of $\mathcal{B}$ is a vector obtained by fixing all indices of $\mathcal{B}$ but one. A *vectorization* of $\mathcal{B}$, denoted by $vec(\mathcal{B})$, is obtained by stacking all mode-1 fibers of $\mathcal{B}$. The mode-d *matricization* of $\mathcal{B}$, denoted by $\boldsymbol{B}_{(d)}$, is a matrix whose columns are mode-d fibers of $\mathcal{B}$ in the lexicographical order. The mode-d product of a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ with a matrix $A \in \mathbb{R}^{J \times I_d}$, denoted by $(\mathcal{B} \times_d A)$, is a tensor whose component is $(\mathcal{B} \times_d A)_{i_1,\ldots,i_{d-1},j_d,i_{d+1}\ldots,i_D} = \sum_{i_d=1}^{I_d} b_{i_1,i_2,\ldots,i_D} a_{j,i_d}$. The *inner product* of two tensors $\mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_D}, \mathcal{S} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ is denoted by $\langle \mathcal{B}, \mathcal{S} \rangle = \sum_{i_1,\ldots,i_D} b_{i_1,\ldots,i_D} s_{i_1,\ldots,i_D}$. A rank-one tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ can be represented by outer products of vectors, i.e., $\mathcal{B} = \boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D$, where $\boldsymbol{b}_d$ is an I_d-dimension vector and "$\circ$" is the *outer product* operator. The *Kronecker product* of two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{B} \in \mathbb{R}^{p \times q}$,

denoted by $\boldsymbol{A} \otimes \boldsymbol{B}$ is an $mp \times nq$ block matrix defined by

$$M = \begin{pmatrix} a_{11}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{pmatrix}.$$

The *Khatri-Rao* product of two matrices $\boldsymbol{A} \in \mathbb{R}^{m\times r}, \boldsymbol{B} \in \mathbb{R}^{p\times r}$, denoted by $\boldsymbol{A} \odot \boldsymbol{B}$, is a $mp \times r$ matrix defined by $[\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \;\; \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \;\; \cdots \;\; \boldsymbol{a}_r \otimes \boldsymbol{b}_r]$, where $\boldsymbol{a}_i \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{b}_i \in \mathbb{R}^{p\times 1}$ for $i = 1, \dots, r$.

7.3 Prognostic modeling using degradation tensors

This chapter considers engineering systems with degradation process that can be represented by tensors, e.g., degradation image streams or profiles. The underlying premise of our prognostic modeling framework rests on using LLS regression to model TTF as a function of degradation tensors. One of the main challenges in fitting such regression models is the high-dimensionality of data which makes coefficients estimation intractable. To address this issue, we use the fact that the essential information of high-dimensional data is often embedded in a low-dimensional subspace. Specifically, we project degradation and coefficient tensors onto a low-dimensional tensor subspace that preserves their inherent information.

To further reduce the number of estimated parameters, coefficient tensors are decomposed using two widely used tensor decomposition techniques, CP and Tucker. The CP decomposition expresses a high-dimensional coefficient tensor as a product of several smaller sized basis matrices [118]. Tucker decomposition, however, expresses a high-dimensional coefficient tensor as a product of a low-dimensional core tensor and several factor matrices [119]. Thus, instead of estimating the coefficient tensor, we only need to estimate its corresponding core tensors and factor/basis matrices, which significantly helps reduce the computational complexity and the required sample for estimation. The parameters of the reduced LLS regression model are estimated using the maximum likelihood (ML) ap-

proach. To obtain the ML estimates, we propose optimization algorithms for CP-based and Tucker-based methods . The optimization algorithms are based on the block relaxation method [120, 121], which alternately updates one block of the parameters while keeping other parameters fixed. Finally, the estimated LLS regression is used to predict and update the RUL of a functioning system. In the following, the details of the proposed methodology is presented.

Our framework is applicable in settings that have a historical dataset of degradation image streams (i.e., degradation tensor) for a sample of units with corresponding TTFs. Let N denote the number of units that make up the historical (training) sample. Let $\mathcal{S}_i \in \mathbb{R}^{q_1 \times q_2 \times q_3}$, for $i = 1, \ldots, n$, denote the degradation tensor and $\tilde{y}_i$ represent the TTF. The following LLS regression model expresses the TTF as a function of a degradation tensor:

$$y_i = \alpha + \langle \mathcal{B}, \mathcal{S}_i \rangle + \sigma \epsilon_i \tag{7.1}$$

where $y_i = \tilde{y}_i$ for a location-scale model and $y_i = \ln(\tilde{y}_i)$ for a log-location-scale model, the scalar α is the intercept of the regression model, and $\mathcal{B} \in \mathbb{R}^{q_1 \times q_2 \times q_3}$ is the tensor of regression coefficients. $\alpha + \langle \mathcal{B}, \mathcal{S}_i \rangle$ is known as the location parameter and σ is the scale parameter. Similar to common LLS regression models [69], we assume that only the location parameter is a function of the covariates, i.e., the degradation tensor. The term ϵ_i is the random noise term with a standard location-scale density $f(\epsilon)$. For example, $f(\epsilon) = \exp(\epsilon - \exp(\epsilon))$ for SEV distribution, $f(\epsilon) = \exp(\epsilon)/(1 + \exp(\epsilon))^2$ for logistic distribution, and $f(\epsilon) = 1/\sqrt{2\pi} \exp(-\epsilon^2/2)$ for normal distribution. Consequently, y_i has a density in the form of $\frac{1}{\sigma} f\left(\frac{y_i - \alpha - \langle \mathcal{B}, \mathcal{S}_i \rangle}{\sigma}\right)$.

The number of parameters in Model (7.1) is given by $2 + \prod_{d=1}^{3} q_d$. Recall that q_d represents the dimension of the d-mode of $\mathcal{B}$. If we consider a simple example of an image stream constituting 100 images of size 40×50, i.e., $\mathcal{S}_i$ is a 3-order tensor in $\mathbb{R}^{40 \times 50 \times 100}$, the number of parameters to be estimated will be quite large: $200,002 = 2 + 40 \times 50 \times 100$. To reduce the number of parameters, as mentioned earlier, we project the degradation tensors

and the coefficient tensor onto a low-dimensional tensor subspace that captures the relevant information of the degradation tensors. The following proposition shows that by using multilinear dimension reduction techniques, we can significantly reduce the dimensions of the coefficient tensor without significant loss of information.

Proposition 4 *Suppose* $\{\mathcal{S}_i\}_{i=1}^n$ *can be expanded by* $\mathcal{S}_i = \tilde{\mathcal{S}}_i \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$*, where* $\tilde{\mathcal{S}}_i \in \mathbb{R}^{p_1 \times p_2 \times p_3}$ *is a low-dimensional tensor and matrices* $\boldsymbol{U}_d \in \mathbb{R}^{p_d \times q_d}$*,* $\boldsymbol{U}_d^\top \boldsymbol{U}_d = \boldsymbol{I}_{q_d}$*,* $p_d < q_d$*,* $d = 1, 2, 3$*. If the coefficient tensor,* $\mathcal{B}$*, is projected onto the tensor subspace spanned by* $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3\}$*, i.e.,* $\tilde{\mathcal{B}} = \mathcal{B} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top$*, where* $\tilde{\mathcal{B}}$ *is the projected coefficient tensor, then* $\langle \mathcal{B}, \mathcal{S}_i \rangle = \langle \tilde{\mathcal{B}}, \tilde{\mathcal{S}}_i \rangle$*.*

The proof of Proposition 4 is given in Appendix F.1. Proposition 4 implies that the original high-dimensional tensors, (i.e., $\mathcal{B}$ and $\mathcal{S}$) and their corresponding low-rank projections (i.e., $\tilde{\mathcal{B}}$ and $\tilde{\mathcal{S}}_i$) result in similar estimates of the location parameter. Using Proposition 4, we can re-express Equation (7.1) as follows:

$$y_i = \alpha + \langle \tilde{\mathcal{B}}, \tilde{\mathcal{S}}_i \rangle + \sigma \epsilon_i. \tag{7.2}$$

The low-dimensional tensor space defined by factor matrices $\boldsymbol{U}_d \in \mathbb{R}^{p_d \times q_d}$ can be obtained by applying multilinear dimension reduction techniques, such as multilinear principal component analysis (MPCA) [21], on the training degradation tensor, $\{\mathcal{S}_i\}_{i=1}^n$. The objective of MPCA is to find a set of orthogonal factor matrices $\{\boldsymbol{U}_d \in \mathbb{R}^{p_d \times q_d}, \boldsymbol{U}_d^\top \boldsymbol{U}_d = \boldsymbol{I}_{q_d}, p_d < q_d\}_{d=1}^3$ such that the projected low-dimensional tensor captures most of the variation in the original tensor. Mathematically, this can be formalized into the following optimization problem:

$$\{\hat{\boldsymbol{U}}_1, \hat{\boldsymbol{U}}_2, \hat{\boldsymbol{U}}_3\} = \underset{\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3}{\arg\max} \sum_{i=1}^n \left\| (\mathcal{S}_i - \bar{\mathcal{S}}) \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top \right\|_F^2 \tag{7.3}$$

where $\bar{\mathcal{S}} = \frac{1}{n} \sum_{i=1}^n \mathcal{S}_i$ is the mean tensor. This optimization problem can be solved iteratively using the algorithm given in Appendix F.2. Additional details regarding the algorithm

and the methods used to determine the dimensionality of the tensor subspace, $\{p_d\}_{d=1}^3$, can be found in [21]. It should be pointed out that if the image stream is noiseless and embedded in a low-dimensional subspace meaning that the corresponding tensor is low rank, then multilinear dimension reduction techniques will not result in any information loss. However, in practice, image streams often contain noise, so multilinear dimension reduction techniques will cause some information loss. However, the lost information is mainly associated with the noise, which is not important in predicting RUL. Multilinear dimension reduction techniques help to reduce the number of parameters to be estimated from $2 + \prod_{d=1}^3 q_d$ in Equation (7.1) to $2 + \prod_{d=1}^3 p_d$ in Equation (7.2) where $2 + \prod_{d=1}^3 p_d << 2 + \prod_{d=1}^3 q_d$.

However, often, the number of reduced parameters (i.e., $2 + \prod_{d=1}^3 p_d$) is still so large that requires further dimension reduction. For example, for a $40 \times 50 \times 100$ tensor, if $p_1 = p_2 = p_3 = 10$, the number of parameters is reduced from 200,002 to 1,002. To further reduce the number of parameters so that they can be estimated by using a limited training sample, we utilize two well-known tensor decomposition techniques namely, CP and Tucker decompositions. We briefly review these decompositions in Sections 7.3.1 and 7.3.2, and discuss how they are incorporated into our prognostic framework.

7.3.1 Dimension reduction via CP decomposition

In CP decomposition, the coefficient tensor $\tilde{\mathcal{B}}$ in Equation (7.2) is decomposed into a sum product of a set of rank one vectors. Given the rank of $\tilde{\mathcal{B}}$, which we denote by k, we have the following decomposition,

$$\tilde{\mathcal{B}} = \sum_{r=1}^{k} \tilde{\boldsymbol{\beta}}_1^{(r)} \circ \tilde{\boldsymbol{\beta}}_2^{(r)} \circ \tilde{\boldsymbol{\beta}}_3^{(r)}, \tag{7.4}$$

where $\tilde{\boldsymbol{\beta}}_d^{(r)} = \left[\tilde{\beta}_{d,1}^{(r)}, \ldots, \tilde{\beta}_{d,p_d}^{(r)}\right]^\top \in \mathbb{R}^{p_d}$, and "$\circ$" denotes the outer product operator. It can be easily shown that $vec(\tilde{\mathcal{B}}) = (\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2 \odot \tilde{\boldsymbol{B}}_1)\mathbf{1}_k$, where $\tilde{\boldsymbol{B}}_d = \left[\tilde{\boldsymbol{\beta}}_d^{(1)}, \ldots, \tilde{\boldsymbol{\beta}}_d^{(k)}\right] \in \mathbb{R}^{p_d \times k}$ for $d = 1, 2, 3$ and $\mathbf{1}_k \in \mathbb{R}^k$ is an k-dimensional vector of ones. Thus, Equation (7.2) can

be re-expressed as follows:

$$\begin{aligned} y_i &= \alpha + \left\langle vec(\tilde{\mathcal{B}}), vec(\tilde{\mathcal{S}}_i) \right\rangle + \sigma\epsilon_i \\ &= \alpha + \left\langle (\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2 \odot \tilde{\boldsymbol{B}}_1)\mathbf{1}_k, vec(\tilde{\mathcal{S}}_i) \right\rangle + \sigma\epsilon_i \end{aligned} \tag{7.5}$$

The number of parameters in Equation (7.5) is $2 + \sum_{d=1}^{3} p_d \times k$, which is significantly smaller than $2 + \prod_{d=1}^{3} p_d$ from (7.2). In our 3-order tensor example, if $p_1 = p_2 = p_3 = 10$ and the rank $k = 2$, the number of parameters decreases from $1,002$ to $62 = 2 + 10 \times 2 + 10 \times 2 + 10 \times 2$.

Parameter Estimation for CP Decomposition

To estimate the parameters of Equation (7.5) using MLE, we maximize the corresponding penalized log-likelihood function:

$$\begin{aligned} &\arg\max_{\boldsymbol{\theta}} \left\{ \ell(\boldsymbol{\theta}) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\} \\ &= \arg\max_{\boldsymbol{\theta}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \left\langle (\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2 \odot \tilde{\boldsymbol{B}}_1)\mathbf{1}_k, vec(\tilde{\mathcal{S}}_i) \right\rangle}{\sigma} \right) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\} \end{aligned} \tag{7.6}$$

where $\boldsymbol{\theta} = (\alpha, \sigma, \tilde{\boldsymbol{B}}_1, \tilde{\boldsymbol{B}}_2, \tilde{\boldsymbol{B}}_3)$ and $r(\tilde{\boldsymbol{B}}_d) = \lambda_d \sum_{r=1}^{k} \sum_{q=1}^{p_d} \|\tilde{\beta}_{d,q}^{(r)}\|_1$. The ℓ_1-norm penalty term encourages the sparsity of $\tilde{\mathcal{B}}$, which helps avoid over-fitting.

The block relaxation method proposed by [120, 121] is used to maximize expression (7.6). Specifically, we iteratively update a block of parameters, say $(\tilde{\boldsymbol{B}}_d, \sigma, \alpha)$, while keeping other components $\{\tilde{\boldsymbol{B}}_{\neq d}\}$ fixed. In each update, the optimization criterion is reduced

to $\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{\ell(\boldsymbol{\theta})-r(\tilde{\boldsymbol{B}}_d)\right\}$.

Next, we show in Proposition 5 that the optimization problem for each block $\tilde{\boldsymbol{B}}_d$ is equivalent to optimizing the penalized log-likelihood function for $y_i=\alpha+\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\rangle+\sigma\epsilon_i$, where $\tilde{\boldsymbol{B}}_d$ is the parameter matrix; $\boldsymbol{X}_{d,i}$ is the predictor matrix defined as follows:

$$\boldsymbol{X}_{d,i}=\begin{cases}\tilde{\boldsymbol{S}}_{i(1)}(\tilde{\boldsymbol{B}}_3\odot\tilde{\boldsymbol{B}}_2), & \text{if } d=1\\ \tilde{\boldsymbol{S}}_{i(2)}(\tilde{\boldsymbol{B}}_3\odot\tilde{\boldsymbol{B}}_1), & \text{if } d=2\\ \tilde{\boldsymbol{S}}_{i(3)}(\tilde{\boldsymbol{B}}_2\odot\tilde{\boldsymbol{B}}_1), & \text{if } d=3\end{cases}, \tag{7.7}$$

where $\tilde{\boldsymbol{S}}_{i(d)}$ is the mode-*d* matricization of $\tilde{\mathcal{S}}_i$ (defined in the Preliminaries Section).

Proposition 5 *Consider the optimization problem in (7.6), given other parameters except* $(\tilde{\boldsymbol{B}}_d,\sigma,\alpha)$*, the optimization problem can be reduced to*

$$\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{-n\ln\sigma+\sum_{i=1}^{n}\ln f\left(\frac{y_i-\alpha-\left\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right)-r(\tilde{\boldsymbol{B}}_d)\right\}. \tag{7.8}$$

The proof of Proposition 5 is provided in Appendix F.3. As pointed out by [122], the estimates of $\alpha,\tilde{\boldsymbol{B}}_d,\sigma$ in optimizing problem (7.8) are not invariant under scaling of the response. To be specific, consider the transformation $y_i'=by_i,\alpha'=b\alpha,\tilde{\boldsymbol{B}}_d'=b\tilde{\boldsymbol{B}}_d,\sigma'=b\sigma$ where $b>0$. Clearly, this transformation does not affect the regression model $y_i=\alpha+\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\rangle+\sigma\epsilon_i$. Therefore, invariant estimates based on the transformed data $(y_i',\boldsymbol{X}_{d,i})$, should satisfy $\hat{\alpha}'=b\hat{\alpha},\hat{\tilde{\boldsymbol{B}}}_d'=b\hat{\tilde{\boldsymbol{B}}}_d,\hat{\sigma}'=b\hat{\sigma}$, where $\hat{\alpha},\hat{\tilde{\boldsymbol{B}}}_d,\hat{\sigma}$ are estimates based on original data $(y_i,\boldsymbol{X}_{d,i})$. However, this does not hold for the estimates obtained by optimizing (7.8). To address this issue, expression (7.8) is modified by dividing the penalty term by the scale

parameter σ:

$$\underset{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}{\arg\max}\left\{-n\ln\sigma+\sum_{i=1}^{n}\ln f\left(\frac{y_i-\alpha-\left\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right)-r(\frac{\tilde{\boldsymbol{B}}_d}{\sigma})\right\}. \tag{7.9}$$

We can show that the resulting estimates possess the invariant property (see Appendix F.4). Note that in the modified problem, the penalty term penalizes the ℓ_1-norm of the coefficients and the scale parameter σ simultaneously, which has some close relations to the Bayesian Lasso [123, 122]. The log-likelihood function in (7.9) is not concave which causes computational problems. We use the following re-parameterization to transform the optimization function to a concave function: $\alpha_0=\alpha/\sigma, \boldsymbol{A}_d=\tilde{\boldsymbol{B}}_d/\sigma, \rho=1/\sigma$. Consequently, the optimization problem can be rewritten as:

$$\underset{\boldsymbol{A}_d,\rho,\alpha_0}{\arg\max}\left\{n\ln\rho+\sum_{i=1}^{n}\ln f\left(\rho y_i-\alpha_0-\langle\boldsymbol{A}_d,\boldsymbol{X}_{d,i}\rangle\right)-r(\boldsymbol{A}_d)\right\}. \tag{7.10}$$

The optimization problem in (7.10) is concave if function $f(\cdot)$ is log-concave, which is the case for most LLS distributions including *normal*, *logistic*, *SEV*, *generalized log-gamma*, *log-inverse Gaussian* [69]. *Lognormal*, *log-logistic* and *Weibull* distributions whose density function is not log-concave can easily be transformed to *normal*, *logistic* and *SEV* distributions, respectively, by taking the logarithm of the TTF. Various optimization algorithms such as coordinate descent [124] and gradient descent [125] can be used for solving (7.10). Algorithm 1 shows the steps of the block relaxation method for optimizing (7.10) and finding the ML estimates of the parameters.

The convergence criterion is defined by $\ell(\tilde{\boldsymbol{\theta}}^{(j+1)})-\ell(\tilde{\boldsymbol{\theta}}^{(j)})<\epsilon$, in which $\ell(\tilde{\boldsymbol{\theta}}^{(j)})$, is

Algorithm 1: Block relaxation algorithm for solving problem (7.6).

Input: $\{\tilde{\mathcal{S}}_i, y_i\}_{i=1}^n$ and rank k

Initialization: Matrices $\tilde{\boldsymbol{B}}_2^{(0)}, \tilde{\boldsymbol{B}}_3^{(0)}$ are initialized randomly.

while convergence criterion not met **do**

for $d = 1, \ldots, 3$ **do**

$$\boldsymbol{X}_{d,i}^{(j+1)} = \begin{cases} \tilde{\boldsymbol{S}}_{i(1)}(\tilde{\boldsymbol{B}}_3^{(j)} \odot \tilde{\boldsymbol{B}}_2^{(j)}), & \text{if } d = 1 \\ \tilde{\boldsymbol{S}}_{i(2)}(\tilde{\boldsymbol{B}}_3^{(j)} \odot \tilde{\boldsymbol{B}}_1^{(j+1)}), & \text{if } d = 2 \\ \tilde{\boldsymbol{S}}_{i(3)}(\tilde{\boldsymbol{B}}_2^{(j+1)} \odot \tilde{\boldsymbol{B}}_1^{(j+1)}), & \text{if } d = 3 \end{cases}$$

$$\boldsymbol{A}_d^{(j+1)}, \rho^{(j+1)}, \alpha_0^{(j+1)} = \underset{\boldsymbol{A}_d, \rho, \alpha_0}{\arg\max} \{n \ln \rho + \sum_{i=1}^n \ln f(\rho y_i - \alpha_0 - \langle \boldsymbol{A}_d, \boldsymbol{X}_{d,i}^{(j+1)} \rangle) - r(\boldsymbol{A}_d)\}$$

$$\tilde{\boldsymbol{B}}_d^{(j+1)} = \boldsymbol{A}_d^{(j+1)} / \rho^{(j+1)}$$

end for

Let $j := j + 1$

end while

Output:$\alpha = \alpha_0/\rho, \sigma = 1/\rho, \{\tilde{\boldsymbol{B}}_d\}_{d=1}^3$

defined as follows:

$$\ell(\tilde{\boldsymbol{\theta}}^{(j)}) = -n \ln \sigma^{(j)} + \sum_{i=1}^n \ln f \left(\frac{y_i - \alpha^{(j)} - \left\langle (\tilde{\boldsymbol{B}}_3^{(j)} \odot \tilde{\boldsymbol{B}}_2^{(j)} \odot \tilde{\boldsymbol{B}}_1^{(j)}) \mathbf{1}_k, vec(\tilde{\mathcal{S}}_i) \right\rangle}{\sigma^{(j)}} \right) - \sum_{d=1}^3 r \left(\frac{\tilde{\boldsymbol{B}}_d^{(j)}}{\sigma^{(j)}} \right) \tag{7.11}$$

where $\tilde{\boldsymbol{\theta}}^{(j)} = (\alpha^{(j)}, \sigma^{(j)}, \tilde{\boldsymbol{B}}_1^{(j)}, \tilde{\boldsymbol{B}}_2^{(j)}, \tilde{\boldsymbol{B}}_3^{(j)})$.

It can be shown that Algorithm 1 exhibits the global convergence property (see Proposition 1 in [126]). In other words, it will converge to a stationary point for any initial point. Since a stationary point is only guaranteed to be a local maximum or saddle point, the algorithm is run several times with different initializations while recording the best results.

Algorithm 1 requires the rank of $\tilde{\mathcal{B}}$ to be known in advance for CP decomposition. In this chapter, the Bayesian information criterion (BIC) is used to determine the appropriate rank. The BIC is defined as $-2\ell(\tilde{\boldsymbol{\theta}}) + m \ln(n)$, where ℓ is the log-likelihood value defined in Equation (7.11), n is the sample size (number of systems) and m is the number of effective

parameters. Here, $m = k(\sum_{d=1}^{3} p_d - 2)$, where $k(-2)$ is used for the scaling indeterminacy in the CP decomposition [23].

7.3.2 Dimension reduction via Tucker decomposition

Tucker decomposition is the second tensor decomposition approach used in this chapter. It is used to reduce the dimensionality of $\tilde{\mathcal{B}}$ as a product of a low-dimensional core tensor and a set of factor matrices as follows:

$$\tilde{\mathcal{B}} = \tilde{\mathcal{G}} \times_1 \tilde{\boldsymbol{B}}_1 \times_2 \tilde{\boldsymbol{B}}_2 \times_3 \tilde{\boldsymbol{B}}_3 = \sum_{r_1=1}^{k_1} \sum_{r_2=1}^{k_2} \sum_{r_3=1}^{k_3} \tilde{g}_{r_1,r_2,r_3} \tilde{\boldsymbol{\beta}}_1^{(r_1)} \circ \tilde{\boldsymbol{\beta}}_2^{(r_2)} \circ \tilde{\boldsymbol{\beta}}_3^{(r_3)}, \tag{7.12}$$

where $\tilde{\mathcal{G}} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ is the core tensor with the element $(\tilde{\mathcal{G}})_{r_1,r_2,r_3} = \tilde{g}_{r_1,r_2,r_3}$, $\tilde{\boldsymbol{B}}_d = \left[\tilde{\boldsymbol{\beta}}_d^{(1)}, \ldots, \tilde{\boldsymbol{\beta}}_d^{(k_d)}\right] \in \mathbb{R}^{p_d \times k_d}$ for $d = 1, 2, 3$ is the factor matrix, "$\times_d$" is the mode-d product operator, and "$\circ$" is the outer product operator. Using this decomposition, Equation (7.2) can be re-expressed as follows:

$$\begin{aligned} y_i &= \alpha + \langle \tilde{\mathcal{B}}, \tilde{\mathcal{S}}_i \rangle + \sigma \epsilon_i \\ &= \alpha + \langle \tilde{\mathcal{G}} \times_1 \tilde{\boldsymbol{B}}_1 \times_2 \tilde{\boldsymbol{B}}_2 \times_3 \tilde{\boldsymbol{B}}_3, \tilde{\mathcal{S}}_i \rangle + \sigma \epsilon_i \end{aligned} \tag{7.13}$$

Parameter Estimation for Tucker Decomposition

The following penalized log-likelihood function is used to compute the ML estimates of the parameters in expression (7.13).

$$\arg\max_{\boldsymbol{\theta}} \left\{ \ell(\boldsymbol{\theta}) - r(\tilde{\mathcal{G}}) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\}$$

$$= \arg\max_{\boldsymbol{\theta}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \left\langle \tilde{\mathcal{G}} \times_1 \tilde{\boldsymbol{B}}_1 \times_2 \tilde{\boldsymbol{B}}_2 \times_3 \tilde{\boldsymbol{B}}_3, \tilde{\mathcal{S}}_i \right\rangle}{\sigma} \right) \right.$$
$$\left. - r(\tilde{\mathcal{G}}) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\}, \tag{7.14}$$

where $\boldsymbol{\theta} = (\alpha, \sigma, \tilde{\mathcal{G}}, \tilde{\boldsymbol{B}}_1, \tilde{\boldsymbol{B}}_2, \tilde{\boldsymbol{B}}_3)$, $r(\tilde{\mathcal{G}}) = \lambda \sum_{r_1=1}^{k_1} \sum_{r_2=1}^{k_2} \sum_{r_3=1}^{k_3} \|\tilde{g}_{r_1,r_2,r_3}\|_1$ and $r(\tilde{\boldsymbol{B}}_d) = \lambda_d \sum_{r_d=1}^{k_d} \sum_{q=1}^{p_d} \|\tilde{\beta}_{d,q}^{(r_d)}\|_1$.

Similar to the CP decomposition model, the block relaxation method is used to solve expression (7.14). To update the core tensor $\tilde{\mathcal{G}}$ given all the factor matrices, the optimization criterion is reduced to $\arg\max_{\tilde{\mathcal{G}}} \left\{ \ell(\boldsymbol{\theta}) - r(\tilde{\mathcal{G}}) \right\}$. Proposition 3 shows that this optimization problem is equivalent to optimizing the penalized log-likelihood function of $y_i = \alpha + \langle vec(\tilde{\mathcal{G}}), \boldsymbol{x}_i \rangle + \sigma\epsilon_i$, where $vec(\tilde{\mathcal{G}})$ is the parameter vector and $\boldsymbol{x}_i$ is the predictor vector defined by $\boldsymbol{x}_i = (\tilde{\boldsymbol{B}}_3 \otimes \tilde{\boldsymbol{B}}_2 \otimes \tilde{\boldsymbol{B}}_1)^\top vec(\tilde{\mathcal{S}}_i)$.

Proposition 6 *Consider the optimization problem in (7.14), given $\{\tilde{\boldsymbol{B}}_1, \tilde{\boldsymbol{B}}_2, \tilde{\boldsymbol{B}}_3\}$, the optimization problem is reduced to*

$$\arg\max_{\tilde{\mathcal{G}}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \left\langle vec(\tilde{\mathcal{G}}), (\tilde{\boldsymbol{B}}_3 \otimes \tilde{\boldsymbol{B}}_2 \otimes \tilde{\boldsymbol{B}}_1)^\top vec(\tilde{\mathcal{S}}_i) \right\rangle}{\sigma} \right) - r(\tilde{\mathcal{G}}) \right\}, \tag{7.15}$$

The proof of Proposition 3 is given in Appendix F.5. To guarantee the invariance property of the estimates and concavity of the optimization function, we apply the following reparameterization: $\rho = 1/\sigma, \alpha_0 = \alpha/\sigma, \mathcal{C} = \tilde{\mathcal{G}}/\sigma, r(\mathcal{C}) = \lambda \sum_{r_1=1}^{k_1} \sum_{r_2=1}^{k_2} \sum_{r_3=1}^{k_3} \frac{\|\tilde{g}_{r_1,r_2,r_3}\|_1}{\sigma}$. This enables us to re-express criterion (7.15) as follows:

$$\arg\max_{\mathcal{C},\rho,\alpha_0} \left\{ n \ln \rho + \sum_{i=1}^{n} \ln f \left(\rho y_i - \alpha_0 - \left\langle vec(\mathcal{C}), (\tilde{\boldsymbol{B}}_3 \otimes \tilde{\boldsymbol{B}}_2 \otimes \tilde{\boldsymbol{B}}_1)^\top vec(\tilde{\mathcal{S}}_i) \right\rangle \right) - r(\mathcal{C}) \right\}. \tag{7.16}$$

To update the factor matrix $\tilde{\boldsymbol{B}}_d$, we fix the core tensor $\tilde{\mathcal{G}}$ and the rest of the factor matrices $\{\tilde{\boldsymbol{B}}_{\neq d}\}$, and maximize the following criterion $\arg\max_{\tilde{\boldsymbol{B}}_d}\left\{\ell(\boldsymbol{\theta}) - r(\tilde{\boldsymbol{B}}_d)\right\}$. Proposition 7 shows that this optimization problem is equivalent to optimizing the log-likelihood function of $y_i = \alpha + \langle \tilde{\boldsymbol{B}}_d, \boldsymbol{X}_{d,i}\rangle + \sigma\epsilon_i$, where $\tilde{\boldsymbol{B}}_d$ is the parameter matrix; $\boldsymbol{X}_{d,i}$ is the predictor matrix defined as follows:

$$\boldsymbol{X}_{d,i} = \begin{cases} \tilde{\boldsymbol{S}}_{i(1)}(\tilde{\boldsymbol{B}}_3 \otimes \tilde{\boldsymbol{B}}_2)\tilde{\boldsymbol{G}}_{(1)}^{\top}, & \text{if } d = 1 \\ \tilde{\boldsymbol{S}}_{i(2)}(\tilde{\boldsymbol{B}}_3 \otimes \tilde{\boldsymbol{B}}_1)\tilde{\boldsymbol{G}}_{(2)}^{\top}, & \text{if } d = 2 \\ \tilde{\boldsymbol{S}}_{i(3)}(\tilde{\boldsymbol{B}}_2 \otimes \tilde{\boldsymbol{B}}_1)\tilde{\boldsymbol{G}}_{(3)}^{\top}, & \text{if } d = 3 \end{cases}, \tag{7.17}$$

where $\tilde{\boldsymbol{S}}_{i(d)}$ and $\tilde{\boldsymbol{G}}_{(d)}$ are the mode-*d* matricization of $\tilde{\mathcal{S}}_i$ and $\tilde{\mathcal{G}}$, respectively.

Proposition 7 *Consider the problem in (7.14), given $\tilde{\mathcal{G}}$ and $\{\tilde{\boldsymbol{B}}_{\neq d}\}$, the optimization problem is reduced to*

$$\arg\max_{\tilde{\boldsymbol{B}}_d}\left\{-n\ln\sigma + \sum_{i=1}^{n}\ln f\left(\frac{y_i - \alpha - \left\langle \tilde{\boldsymbol{B}}_d, \boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right) - r(\tilde{\boldsymbol{B}}_d)\right\}. \tag{7.18}$$

The proof of Proposition 7 is provided in Appendix F.6. Similar to expression (7.9), we use penalty term $r(\frac{\tilde{\boldsymbol{B}}_d}{\sigma})$ and let $\rho = 1/\sigma, \alpha_0 = \alpha/\sigma, \boldsymbol{A}_d = \tilde{\boldsymbol{B}}_d/\sigma$. Consequently, we obtain the following optimization subproblem for parameter estimation:

$$\arg\max_{\boldsymbol{A}_d}\left\{n\ln\rho + \sum_{i=1}^{n}\ln f\left(\rho y_i - \alpha_0 - \langle \boldsymbol{A}_d, \boldsymbol{X}_{d,i}\rangle\right) - r(\boldsymbol{A}_d)\right\}. \tag{7.19}$$

The pseudocode for the block relaxation algorithm is summarized in Algorithm 2. The

Algorithm 2: Block relaxation algorithm for solving problem (7.13).

Input: $\{\tilde{\mathcal{S}}_i, y_i\}_{i=1}^n$ and rank $\{k_d\}_{d=1}^3$

Initialization: Core tensor $\mathcal{G}^{(0)}$ and matrices $\tilde{\boldsymbol{B}}_2^{(0)}, \tilde{\boldsymbol{B}}_3^{(0)}$ are initialized randomly.

while convergence criterion not met **do**

for $d = 1, \ldots, 3$ **do**

$$\boldsymbol{X}_{d,i}^{(j+1)} = \begin{cases} \tilde{\boldsymbol{S}}_{i(1)}(\tilde{\boldsymbol{B}}_3^{(j)} \otimes \tilde{\boldsymbol{B}}_2^{(j)})\{\tilde{\boldsymbol{G}}_{(1)}^{(j)}\}^\top, & \text{if } d = 1 \\ \tilde{\boldsymbol{S}}_{i(2)}(\tilde{\boldsymbol{B}}_3^{(j)} \otimes \tilde{\boldsymbol{B}}_1^{(j+1)})\{\tilde{\boldsymbol{G}}_{(2)}^{(j)}\}^\top, & \text{if } d = 2 \\ \tilde{\boldsymbol{S}}_{i(3)}(\tilde{\boldsymbol{B}}_2^{(j+1)} \otimes \tilde{\boldsymbol{B}}_1^{(j+1)})\{\tilde{\boldsymbol{G}}_{(3)}^{(j)}\}^\top, & \text{if } d = 3 \end{cases}$$

$$\boldsymbol{A}_d^{(j+1)}, \rho^{(j+1)}, \alpha_0^{(j+1)} = \underset{\boldsymbol{A}_d, \rho, \alpha_0}{\arg\max}\{n \ln \rho + \sum_{i=1}^n \ln f(\rho y_i - \alpha_0 - \langle \boldsymbol{A}_d, \boldsymbol{X}_{d,i}^{(j+1)} \rangle) - r(\boldsymbol{A}_d)\}$$

$$\tilde{\boldsymbol{B}}_d^{(j+1)} = \boldsymbol{A}_d^{(j+1)} / \rho^{(j+1)}$$

end for

$$\mathcal{C}^{(j+1)}, \rho^{(j+1)}, \alpha_0^{(j+1)} = \underset{\mathcal{C}, \rho, \alpha_0}{\arg\max}\{n \ln \rho + \sum_{i=1}^n \ln f(\rho y_i - \alpha_0 - \langle vec(\mathcal{C}), (\tilde{\boldsymbol{B}}_3^{(j+1)} \otimes \tilde{\boldsymbol{B}}_2^{(j+1)} \otimes \tilde{\boldsymbol{B}}_1^{(j+1)})^\top vec(\tilde{\mathcal{S}}_i) \rangle) - r(\mathcal{C})\}$$

$$\mathcal{G}^{(j+1)} = \mathcal{C}^{(j+1)} / \rho^{(j+1)}$$

Let $j := j + 1$

end while

Output: $\alpha = \alpha_0/\rho, \sigma = 1/\rho, \mathcal{G}, \{\tilde{\boldsymbol{B}}_d\}_{d=1}^3$

convergence criterion is defined by $\ell(\tilde{\boldsymbol{\theta}}^{(j+1)}) - \ell(\tilde{\boldsymbol{\theta}}^{(j)}) < \epsilon$, where $\ell(\tilde{\boldsymbol{\theta}}^{(j)})$, is given by

$$\ell(\tilde{\boldsymbol{\theta}}^{(j)}) = -n \ln \sigma^{(j)} + \sum_{i=1}^n \ln f\left(\frac{y_i - \alpha^{(j)} - \left\langle \tilde{\mathcal{G}}^{(j)} \times_1 \tilde{\boldsymbol{B}}_1^{(j)} \times_2 \tilde{\boldsymbol{B}}_2^{(j)} \times_3 \tilde{\boldsymbol{B}}_3^{(j)}, \tilde{\mathcal{S}}_i \right\rangle}{\sigma}\right) - r(\tilde{\mathcal{G}}^{(j)}) - \sum_{d=1}^3 r\left(\tilde{\boldsymbol{B}}_d^{(j)}\right), \tag{7.20}$$

where $\tilde{\boldsymbol{\theta}}^{(j)} = (\alpha^{(j)}, \rho^{(j)}, \mathcal{G}^{(j)}, \tilde{\boldsymbol{B}}_1^{(j)}, \tilde{\boldsymbol{B}}_2^{(j)}, \tilde{\boldsymbol{B}}_3^{(j)})$.

The set of ranks (i.e., k_1, k_2, k_3) used in the Tucker decomposition is an input to Algorithm 2. BIC is also used here to determine the appropriate rank, where ℓ is the log-likelihood value defined in Equation (7.20), n is the sample size (number of systems) and $m = \sum_{d=1}^3 p_d k_d + \prod_{d=1}^3 k_d - \sum_{d=1}^3 k_d^2$ is the number of effective parameters. Here the term $-\sum_{d=1}^3 k_d^2$ is used to adjust for the non-singular transformation indeterminacy in the

Tucker decomposition [23].

Using BIC for rank selection in the Tucker-based tensor regression model can be computationally prohibitive. For example, for a 3-order tensor, there are totally $27 = 3^3$ rank candidates when the maximum rank in each dimensionality is 3. Increasing the maximum rank to 4 and 5, the number of rank candidates is increased to $64 = 4^3$ and $125 = 5^3$, respectively. To address this challenge, we propose a computationally efficient heuristic method that automatically selects an appropriate rank. First, an initial coefficient tensor is estimated by regressing each pixel against the TTF. Next, high-order singular value decomposition (HOSVD) [127] is applied to the estimated tensor. HOSVD works by applying regular SVD to matricizations of the initial tensor on each mode. The rank of each mode can be selected by using FVE [19] and the resulting eigenvector matrix is the factor matrix for that mode. Given the initial tensor and its estimated factor matrices, we can estimate the core tensor. The core tensor and factor matrices estimated by HOSVD are used for initialization in Algorithm 2. As pointed out by various studies in the literature, HOSVD often performs reasonably well as an initialization method for iterative tensor estimation algorithms [117, 21].

7.4 RUL prediction and realtime updating

The goal of this chapter is to predict and update the RUL of partially degraded systems using in-situ degradation image streams. To achieve this, we utilize the LLS regression modeling framework discussed in Section 3, and update the trained model based on data streams observed from fielded systems. The LLS regression model requires that the degradation image streams of the training systems and the fielded system being monitored to have the same dimensionality. In other words, both should have the same number of degradation images or profile length. In reality, this attribute is difficult to maintain for two reasons; (1) different systems have different failure times, and (2) an equipment is typically shutdown after failure and no further observations can be made beyond the failure time.

Assuming the sampling (observation) time intervals are the same for all systems, a system with a longer failure time has more degradation data than a system with a short failure time.

To address this challenge, we adopt the time-varying regression framework used in [19, 128]. The idea of the time-varying regression is that systems whose TTF are shorter than the current observation time (of the fielded system) are excluded from the training dataset. Next, the degradation data of the chosen systems are truncated at the current observation time. By doing this, we ensure that the truncated degradation tensors of the chosen systems and the real-time observed degradation tensors of the fielded system possess the same dimensionality.

We summarize the process of predicting and updating the RUL of a fielded system as follows:

(i) At each sampling time t_n, a new degradation image is observed from a fielded system. Systems whose TTF are longer than t_n are chosen from the training dataset.

(ii) The image streams of the chosen systems are then truncated at time t_n by keeping only the images observed at times $\{t_1, t_2, \ldots, t_n\}$. The truncated image streams constitutes a new "training dataset," hereafter referred to as *truncated training dataset*.

(iii) A dimensionality reduction technique, such as MPCA, is applied to the *truncated training dataset* to obtain a low-dimensional tensor subspace of the *truncated training dataset*. Tensors in the *truncated training dataset* are then projected to the tensor subspace and their low-dimensional approximations are estimated.

(iv) The low-dimensional approximation tensors are used to fit the regression model in Equation (7.2), and the parameters (i.e., $\hat{\alpha}_{t_n}, \hat{\tilde{\mathcal{B}}}_{t_n}$ and $\hat{\sigma}_{t_n}$) are estimated via one of the methods described in Sections 7.3.1 and 7.3.2.

(v) The image stream from the fielded system is projected onto the tensor subspace estimated in step (iii), and its low-dimensional approximation (denoted as $\tilde{\mathcal{S}}_{t_n}$) is also

estimated. Next, the approximated tensor is input into the regression model estimated in step (iv), and the TTF is predicted as follows:

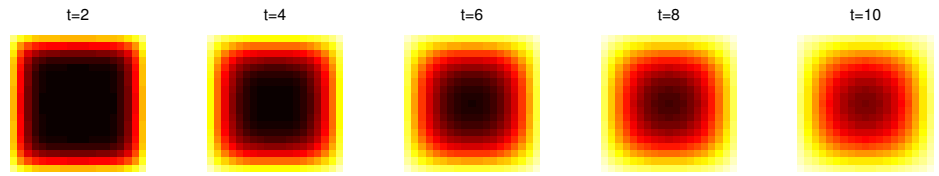
$$\hat{y}_{t_n} = \hat{\alpha}_{t_n} + \langle \hat{\tilde{\mathcal{B}}}_{t_n}, \tilde{\mathcal{S}}_{t_n} \rangle + \hat{\sigma}_{t_n} \epsilon_i, \tag{7.21}$$

which implies the predicted TTF follows an LLS distribution with location parameter $\hat{\alpha}_{t_n} + \langle \hat{\tilde{\mathcal{B}}}_{t_n}, \tilde{\mathcal{S}}_{t_n} \rangle$ and scale parameter $\hat{\sigma}_{t_n}$, i.e., $\hat{y}_{t_n} \sim LLS(\hat{\alpha}_{t_n} + \langle \hat{\tilde{\mathcal{B}}}_{t_n}, \tilde{\mathcal{S}}_{t_n} \rangle, \hat{\sigma}_{t_n})$. From this distribution, we can calculate both the point and interval estimate of the predicted TTF. The RUL is then obtained by subtracting the current observation time from the predicted TTF.

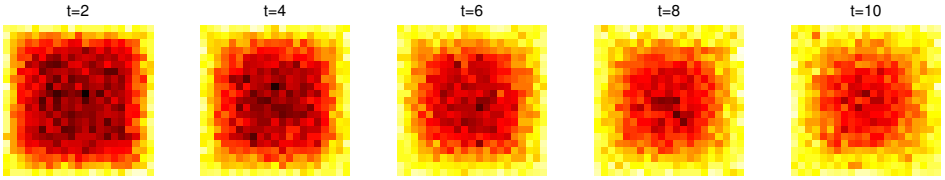
Note that steps (i)-(iv) can be done offline. That is, given a training dataset, we can construct *truncated training datasets* with images observed at time $\{t_1, t_2\}$, $\{t_1, t_2, t_3\}$, $\{t_1, t_2, t_3, t_4\}$, ..., respectively. Regression models are then estimated based on all the possible *truncated training datasets*. Once a new image is observed at say time t_n, the appropriate regression model with images $\{t_1, \ldots, t_n\}$ is chosen, and the RUL of the fielded system is estimated in step (v). This approach enables real-time RUL predictions.

7.5 Numerical study I

In this section, we validate the effectiveness of model and rank selection criteria (BIC and the heuristic method) using simulated degradation image streams. We assume the underlying physical degradation follows a heat transfer process based on which simulated degradation image streams are generated.



(a) Without noise



(b) With noise

Figure 7.2: Simulated degradation images based on heat transfer process.

7.5.1 Data generation

Suppose for system i, the degradation image stream, denoted by $\mathcal{S}_i(x, y, t), i = 1, \ldots, 1000$, is generated from the following heat transfer process:

$$\frac{\partial \mathcal{S}_i(x, y, t)}{\partial t} = \alpha_i \left(\frac{\partial^2 \mathcal{S}_i}{\partial x^2} + \frac{\partial^2 \mathcal{S}_i}{\partial y^2}\right), \tag{7.22}$$

where $(x, y); 0 \leq x, y \leq 0.05$ represents the location of each image pixel, α_i is the thermal diffusivity coefficient for system i and is randomly generated from a uniform distribution $\mathcal{U}(0.5 \times 10^{-5}, 1.5 \times 10^{-5})$ and t is the time frame. The initial and boundary conditions are set such that $\mathcal{S}|_{t=1} = 0$ and $\mathcal{S}|_{x=0} = \mathcal{S}|_{x=0.05} = \mathcal{S}|_{y=0} = \mathcal{S}|_{y=0.05} = 1$. At each time t, the image is recorded at locations $x = \frac{j}{n+1}, y = \frac{k}{n+1}, j, k = 1, \ldots, n$, resulting in an $n \times n$ matrix. Here we set $n = 21$ and $t = 1, \ldots, 10$, which leads to 10 images of size 21×21 for each system represented by a $21 \times 21 \times 10$ tensor. Finally, i.i.d noises $\varepsilon \sim N(0, 0.01)$ are added to each pixel. Example degradation images observed at time $t = 2, 4, 6, 8, 10$ from a simulated system are shown in Figure 7.2, in which (a) and (b) show images without and with noise, respectively.

To simulate the TTF of each system two sets of coefficient tensors are used. The first set,

denoted by $\mathcal{B}_C$, is simulated in the form of basis matrices with rank 2 used in CP decomposition. Specifically, three matrices, i.e., $\boldsymbol{B}_{C,1} \in \mathbb{R}^{21\times 2}, \boldsymbol{B}_{C,2} \in \mathbb{R}^{21\times 2}, \boldsymbol{B}_{C,3} \in \mathbb{R}^{10\times 2}$ are generated. To induce sparsity, we randomly set half of elements of each matrix to be 0. The values of the remaining 50% elements are randomly generated from a uniform distribution $unif(-1, 1)$. The TTF, denoted by $y_{C,i}$, is generated by using $y_{C,i} = \langle vec(\mathcal{B}_C), vec(\mathcal{S}_i)\rangle + \sigma\epsilon_i$, where $vec(\mathcal{B}_C) = (\boldsymbol{B}_{C,3} \odot \boldsymbol{B}_{C,2} \odot \boldsymbol{B}_{C,1})\mathbf{1}_2$, ϵ_i follows a standard smallest extreme value distribution SEV$(0, 1)$ and σ is 5% times the standard deviation of the location parameter, i.e., $\langle vec(\mathcal{B}_C), vec(\mathcal{S}_i)\rangle$.

The second set, denoted by $\mathcal{B}_T$, is simulated in the form of core and factor matrices with rank $(2, 1, 2)$ used in Tucker decomposition. Specifically, a core tensor $\mathcal{G}_T \in \mathbb{R}^{2\times 1\times 2}$ and three factor matrices $\boldsymbol{B}_{T,1} \in \mathbb{R}^{21\times 2}, \boldsymbol{B}_{T,2} \in \mathbb{R}^{21\times 1}, \boldsymbol{B}_{T,3} \in \mathbb{R}^{10\times 2}$ are generated. All the elements of the core tensor $\mathcal{G}_T$ are set to 1. Furthermore, half of elements of matrices $\boldsymbol{B}_{T,1}, \boldsymbol{B}_{T,2}, \boldsymbol{B}_{T,3}$ are randomly set to 0 and the remaining elements are randomly generated from $unif(-1, 1)$. The TTF, $y_{T,i}$, is generated via $y_{T,i} = \langle vec(\mathcal{B}_T), vec(\mathcal{S}_i)\rangle + \sigma\epsilon_i$, where $vec(\mathcal{B}_T) = \mathcal{G}_T \times_1 \boldsymbol{B}_{T,1} \times_2 \boldsymbol{B}_{T,2} \times_3 \boldsymbol{B}_{T,3}$, ϵ_i follows a standard smallest extreme value distribution SEV$(0, 1)$ and σ is 5% times the standard deviation of the location parameter, i.e., $\langle vec(\mathcal{B}_T), vec(\mathcal{S}_i)\rangle$.

In the following subsection, we study the performance of the BIC criterion and our heuristic rank selection method in identifying the correct LLS distribution (i.e., SEV) as well as the right rank. We randomly select 500 of the simulated systems for training and the remaining 500 systems for test.

7.5.2 Model and rank selection

We first apply CP-based tensor regression in Equation (7.5) to the training dataset, $\{y_{C,i}, \mathcal{S}_i\}_{i=1}^{500}$, and use Algorithm 1 to estimate the model parameters for different ranks and for four LLS distributions, namely, *normal, SEV*, *lognormal* and *Weibull*. The BIC value is then computed for each distribution and rank combination as discussed in Section 7.3.1. As pointed

Table 7.1: BIC values for CP-based tensor regression.

Rank	1	2	3	4	5	6	7
SEV	620.5	**-1535.3**	-1383.4	-1232.7	-1122.9	-1014.4	-805.9
Normal	550.0	-1422.6	-1273.6	-1153.2	-1064.7	-1013.2	-1114.0
Weibull	618.1	-643.6	-472.6	-301.9	-180.5	-103.5	-54.0
Lognormal	610.5	-336.3	-187.7	-75.5	9.6	67.4	74.3

out earlier, the block relaxation method in Algorithm 1 only guarantees a local optimum and hence, we shall run the algorithm 10 times using randomized initializations and record the smallest BIC. Here, the randomized initializations are achieved by setting each entry of matrices $\tilde{\boldsymbol{B}}_2^{(0)}$ and $\tilde{\boldsymbol{B}}_3^{(0)}$ with a random number generated from a uniform distribution $\mathcal{U}(-1,1)$. The BIC values for all combinations are reported in Table 7.1. From Table 7.1, it can be seen that the smallest BIC value is -1535.3, which belongs to *SEV* distribution with rank $r = 2$. This coincides with the rank and the distribution we used to generate the data.

Similarly, the Tucker-based tensor regression model in Equation (7.13) is applied to the training dataset, $\{y_{T,i}, \mathcal{S}_i\}_{i=1}^{500}$ and Algorithm 2 (see Section 7.3.2) is used to estimate the parameters. Again, the randomized initializations in Algorithm 2 are achieved by setting each entry of the core tensor $\mathcal{G}$ and matrices $\tilde{\boldsymbol{B}}_2^{(0)}$ and $\tilde{\boldsymbol{B}}_3^{(0)}$ with a random number generated from a uniform distribution $\mathcal{U}(-1,1)$. A total of 27 different rank combinations are tested under four distributions, *normal, SEV*, *lognormal* and *Weibull*. Again, for each distribution-rank combination, Algorithm 2 is run with 10 randomized initializations, and the smallest BIC value is reported in Table 7.2 .

Table 7.2 indicates that the smallest BIC value (-1313.5) is associated with the *SEV* distribution with rank $(2,1,2)$, which again matches the rank and the distribution that was used to generate the data. Therefore, we can conclude that the BIC criterion is effective in selecting an appropriate distribution and the correct rank of the tensors in the LLS regression. In Table 7.3, we also report the results of the heuristic rank selection method for Tucker. It can be seen from Table 7.3 that the heuristic rank selection method selects rank

Table 7.2: BIC values for Tucker-based tensor regression.

Rank	(1,1,1)	(1,1,2)	(1,1,3)	(1,2,1)	(1,2,2)	(1,2,3)	(1,3,1)	(1,3,2)	(1,3,3)
SEV	-163.3	-113.2	-75.8	-44.8	-59.0	-15.7	61.0	52.6	29.6
Normal	-199.0	-149.3	-112.0	-81.0	-82.8	-39.3	24.8	28.9	15.5
Weibull	-73.9	-24.4	13.0	44.1	35.9	79.2	149.6	147.4	133.5
Lognormal	-83.7	-33.9	3.4	34.4	28.6	71.6	140.0	140.2	141.9
Rank	(2,1,1)	(2,1,2)	(2,1,3)	(2,2,1)	(2,2,2)	(2,2,3)	(2,3,1)	(2,3,2)	(2,3,3)
SEV	-44.8	**-1313.5**	-1269.8	-16.1	-1212.7	-1202.9	95.4	-1115.0	-1106.5
Normal	-80.9	-1259.1	-1215.6	-22.9	-1149.7	-1130.8	89.3	-1048.6	-1028.2
Weibull	44.1	-733.8	-690.2	66.1	-633.2	-607.1	178.1	-543.5	-508.1
Lognormal	34.4	-497.8	-454.3	85.2-	-402.7	-394.4	197.2	-306.2	-292.6
Rank	(3,1,1)	(3,1,2)	(3,1,3)	(3,2,1)	(3,2,2)	(3,2,3)	(3,3,1)	(3,3,2)	(3,3,3)
SEV	60.7	-1201.8	-1224.9	95.5	-1156.4	-1164.0	113.0	-1071.4	-1074.4
Normal	24.9	-1147.2	-1153.2	88.8	-1093.2	-1082.6	129.4	-1009.0	-999.2
Weibull	149.7	-621.9	-613.1	177.8	-572.3	-539.2	205.6	-488.5	-468.2
Lognormal	139.9	-385.9	-391.0	197.5	-337.9	-331.4	238.5	-252.0	-262.3

$(1, 1, 1)$ under *normal* and *lognormal* distributions, while selects rank $(2, 2, 2)$ under *SEV* distribution and $(1, 2, 2)$ under *Weibull* distribution. The smallest BIC values (-1212.3) is achieved under *SEV* distribution with rank $(2, 2, 2)$, which is close to the actual rank.

Table 7.3: Distribution and rank selection results by using heuristic rank selection method.

LLS Distribution	Rank	BIC
SEV	(2,2,2)	**-1212.3**
Normal	(1,1,1)	-199.0
Weibull	(1,2,2)	36.1
Lognormal	(1,1,1)	-83.7

7.6 Numerical study II

In this section, we validate the prediction capability of our methodology with the two types of decomposition approaches using simulated degradation image streams.

7.6.1 Data generation

Similar to Section 7.5, we assume the underlying physical degradation follows a heat transfer process based on which simulated degradation image streams are generated. Specifically, the degradation image streams are generated using Equation (7.22). Here, the thermal diffusivity coefficient for system $i, i = 1, \ldots, 500$, is randomly generated from a uniform distribution $\mathcal{U}(5 \times 10^{-5}, 1 \times 10^{-4})$. Each system has 100 images with size 51×51. As a result, each system is represented by a $51 \times 51 \times 100$ tensor. Two types of noise are added to each image. The first type of noise is generated from a spatial Gaussian process, $GP(\mathbf{0}, K(\cdot, \cdot))$. Here, the covariance function $K(\boldsymbol{s}_1, \boldsymbol{s}_2) = \sigma^2 \exp(-\phi \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|)$, where $\sigma^2 = 0.01, \phi = 0.25$, and $\|\boldsymbol{s}_1 - \boldsymbol{s}_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ is the Euclidean distance between pixels $\boldsymbol{s}_1 = (x_1, y_1)$ and $\boldsymbol{s}_2 = (x_2, y_2)$. The second type of noise, independent and identically distributed from a normal distribution $\mathcal{N}(0, 0.02^2)$, is added to each pixel.

In prognostic analysis, a TTF is usually defined as the time that the degradation signal crosses a predefined failure threshold. However, for applications with degradation image streams, it is difficult to define such a failure threshold. To address this challenge, in this simulation, the TTF of system i is generated from $\tilde{y}_i = \alpha_i$, where α_i is the thermal diffusivity coefficient. The motivation behind this is that the thermal diffusivity coefficient α_i controls system i's degradation rate, which determines the TTF of the system in reality. Therefore, it is reasonable to use the diffusivity coefficient as a proxy of TTF.

In the following two subsections, we first introduce the models used to benchmark our proposed methodology. Then, we compare the performance of our method with the benchmarking models in terms of the prediction capability.

7.6.2 Benchmarks and validation settings

Our proposed methods, designated as "CP" and "Tucker," work by first applying MPCA to the degradation image streams to extract their low-dimensional projected tensors. Next, the projected tensors are regressed against TTFs using the CP- and Tucker-based LLS re-

gression model, respectively. We compare the performance of our methodologies with five baseline models. The first two benchmarking models are similar to our proposed methodologies except that they do not apply MPCA on the degradation image streams. We refer to them as "CP (No MPCA)" and "Tucker (No MPCA)," respectively.

The third baseline model, which we designated as "FPCA," uses functional principal components analysis (FPCA) to model the overall image intensity. To be specific, we first transform the degradation image stream of each system into a time-series signal by taking the average intensity of each observed image. Next, FPCA is applied to the time-series signals to extract features. FPCA is a popular functional data analysis technique that identifies the important sources of patterns and variations among functional data (time-series signals in our case)[17]. The time-series signals are projected to a low-dimensional feature space spanned by the eigen-functions of the signals' covariance function and provides fused features called FPC-scores. Finally, FPC-scores are regressed against the TTFs by using LLS regression. More details about this FPCA prognostic model can be found in [19].

The fourth benchmark is referred to as "PCA," which uses principal components analysis (PCA) to model the vectorized image intensity. Specifically, we first transform each image (in $\mathbb{R}^{51\times 51}$) to a vector (in $\mathbb{R}^{1\times 2601}$). Next, we construct a signal matrix (in $\mathbb{R}^{100n_0\times 2601}$, n_0 is the number of training systems), each row of which contains a vectorized image. Third, PCA is applied to the signal matrix to reduce dimensionality and extract features. The extracted features from all the images of a system are concatenated to be that system's covariates, which are then regressed against the TTFs by using LLS regression.

We refer to the last baseline method as "B-spline." This approach is inspired by [104], in which the authors use penalized B-spline to fit the smooth mean function of an image stream. In this simulation, we also use penalized B-spline to fit each degradation image stream, and use the resulting coefficients as that image stream's low-dimensional features. We then regress the features against the TTFs using LLS regression.

We evaluate the performance of our methods and the benchmarks under different train-

ing sample sizes: (i) large and (ii) small, where the image streams of the first 400 and 100 systems in the simulated dataset are used for model training, respectively. The image streams of the last 100 systems in the simulated dataset are used for validation. Prediction errors are calculated using Equation (2.25). In this simulation study, the LLS distributions for the regression models are selected by using BIC. The rank for CP-based models is selected using BIC, and the rank for Tucker-based models is chosen by the heuristic method discussed in Section 7.3.2. The number of principal components for "FPCA" and "PCA" is selected using cross-validation (CV). The order of spline basis, knots number, and regularization tuning parameters are chosen using generalized cross-validation (GCV) following the suggestion of [104].

The simulation scenarios were performed using MATLAB 2012b in a 64-bit Unix system with the Xeon X5560 CPU @2.80 GHz processor and 150.0 GB RAM.

7.6.3 Results and analysis

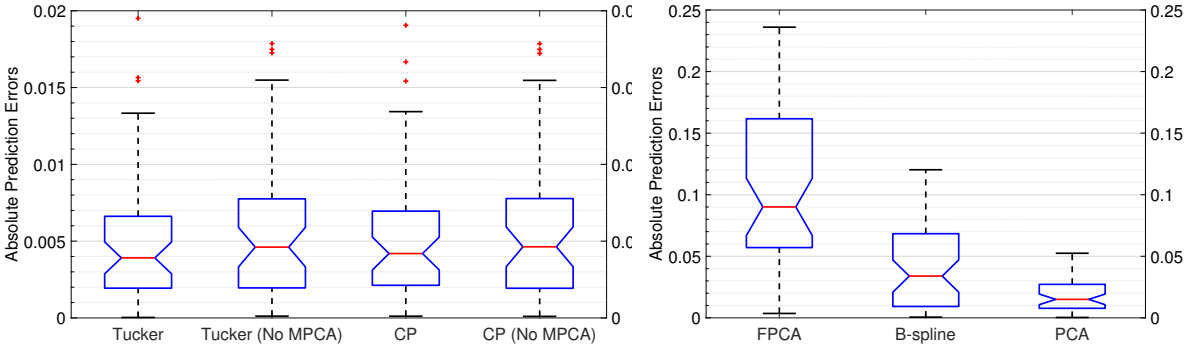
We first evaluate the computational time of our proposed methodologies and the benchmarking models. To do this, we take the image streams from the first 400 systems in the simulated dataset for training and randomly select 1 system from the remaining 100 systems for test. The computational time is reported in Table 7.4, which illustrates that our proposed methodologies are computationally less efficient than "FPCA" but more efficient than "PCA" and "B-spline." In addition, it also indicates that the computational time for "CP" and "Tucker" are similar to the computational time of "CP (No MPCA)" and " Tucker (No MPCA)."

Table 7.4: Computational time (unit: second).

Method	CP	Tucker	CP (No MPCA)	Tucker (No MPCA)	FPCA	PCA	B-spline
Time	118	126	113	154	23	656	232

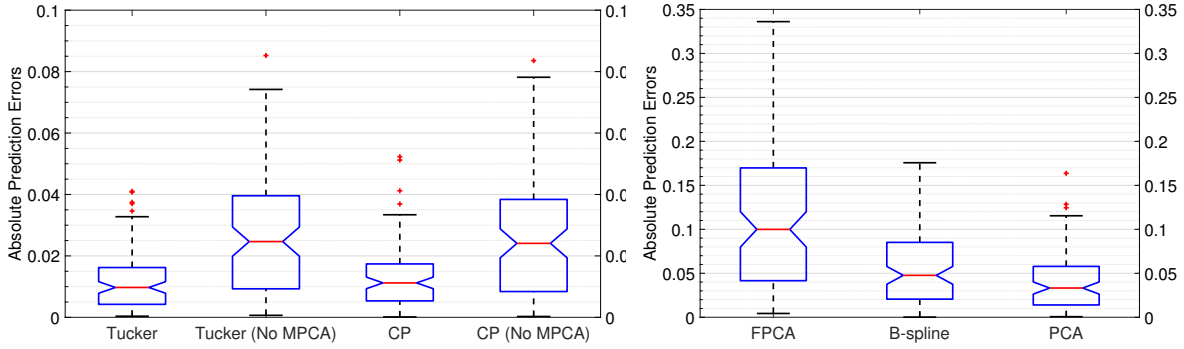
Next, we report box plots of the prediction errors of the different approaches at dif-

ferent training sample sizes in Figures 7.3 and 7.4. In the following subsections, we first evaluate the performance of our proposed tensor-based models by comparing them with non-tensor-based benchmarks. Then, we evaluate the effectiveness of the multilinear dimension reduction technique incorporated in our tensor-based regression models.



(a) Tensor-based models (b) Non-tensor-based models

Figure 7.3: Prediction errors with large training sample size.



(a) Tensor-based models (b) Non-tensor-based models

Figure 7.4: Prediction errors with small training sample size

Performance comparison between tensor-based and non-tensor-based models

Figures 7.3 and 7.4 indicate that, at both large and small training sample size scenarios, our proposed tensor-based regression models (i.e., "CP," "Tucker," "CP (No MPCA)," and "Tucker (No MPCA)") achieved smaller absolute prediction errors (in terms of both

mean and variance) than the non-tensor-based benchmarks (i.e., "FPCA," "PCA," and "B-spline"). For example, in Figure 7.3 (a), the median absolute prediction errors (and the interquartile range, i.e., IQR) of the tensor-based models are around $0.5\%(1.5\%)$, while they are around $9\%(25\%)$, $3.2\%(12\%)$, and $1.5\%(5\%)$ for "FPCA," "PCA," and "B-spline," respectively. Similar phenomenon can also be observed in Figure 7.4. This implies that our tensor-based models outperform other benchmarks in terms of both prediction accuracy and precision. We believe this is because our tensor-based models have the following characteristics: (i) Our methodologies are capable of capturing the spatio-temporal correlation structure in each image stream; (ii) the multilinear dimensionality reduction technique used in our methods is able to maximize the stream-to-stream (system-to-system) variation captured in the low-dimensional projection space; (iii) our methods can be seen as supervised dimension reduction methodologies, which use TTF information to supervise the dimension reduction of the coefficient tensor.

Compared with our models, none of the benchmarking models, i.e., "FPCA," "PCA," or "B-spline," uses TTF information to supervise the dimension reduction process. In addition, the benchmarking models have some other limitations that compromise their prediction capabilities. For example, "FPCA" transforms the degradation image stream of each system into a time-series signal by taking the average intensity of each observed image. This results in a significant loss of spatial information in each image, and thus, compromising the prediction accuracy.

"PCA" first extracts low-dimensional features (i.e., PC-scores) by applying regular linear PCA to the set of vectorized images without considering their temporal dependency. Next, the PC-scores from all the images of each system are concatenated as the system's regression covariates. By doing so, "PCA" hierarchically captures the spatial and temporal correlation among image streams. However, the dimension reduction achieved by"PCA" is insufficient since "PCA" maximizes the image-to-image variation but not the system-to-system variation captured in the low-dimensional projection space. Furthermore, for image

streams containing non-separable spatio-temporal correlation, "PCA" breaks the correlation structure by modeling spatial and temporal correlation separately. In addition, for applications where each system has numerous images, "PCA" may result in a large number of covariates for each system, and thus will pose a parameter estimation challenge (the number of parameters is equal to the number of PC-scores extracted from each image times the number of images in each system).

The "B-spline" benchmark individually fits each image stream using penalized B-spline, and the fitting coefficients are treated as low-dimensional features of that system. "B-spline" is capable of capturing the spatio-temporal correlation in each image stream. However, it fails to consider the stream-to-stream (system-to-system) variation, and thus it is an insufficient dimensionality reduction technique for our prognostic application.

All these aforementioned limitations of the benchmarking models compromise their prediction capabilities, so they did not perform well as our proposed tensor-based regression methodologies.

Performance evaluation of the multilinear dimension reduction methodology

In this subsection, we evaluate the performance of the multilinear dimension reduction technique incorporated in our methodologies.

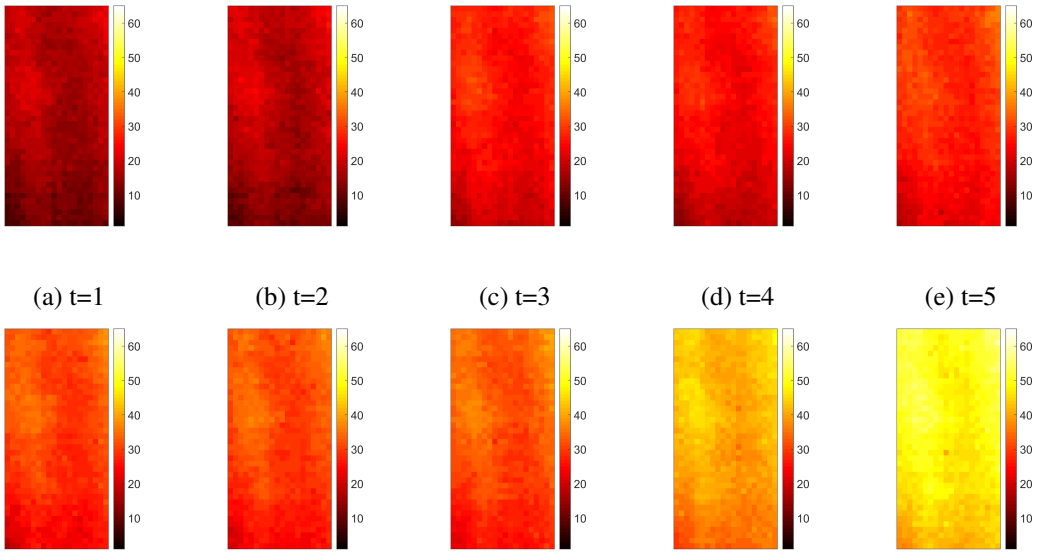
Figure 7.3 (a) illustrates that when the training sample size is large, the prediction errors of "CP" and "CP (No MPCA)" are close to each other. The median (IQR) of the prediction errors for "CP" and "CP (No MPCA)" are around $0.41\%(1.4\%)$ and $0.47\%(1.5\%)$, respectively. A similar phenomenon can also be observed between "Tucker" and "Tucker (No MPCA)," where the median (IQR) of the prediction errors are around $0.4\%(1.3\%)$ and $0.45\%(1.5\%)$, respectively. This implies that, when the training sample size is large enough, the prediction capabilities of our proposed tensor-based regression models are not affected by incorporating the multilinear dimension reduction technique. Figure 7.4 (a) indicates that when the training sample size is small, "CP" and "Tucker" achieve better

prediction results than "CP (No MPCA)" and "Tucker (No MPCA)," respectively. For instance, the median (IQR) of the prediction errors for "CP" and "Tucker" are $1.2\%(3.5\%)$ and $1\%(3.5\%)$, respectively. However, they are $2.5\%(8\%)$ and $2.5\%(7.5\%)$ for "CP (No MPCA)" and "Tucker (No MPCA)" respectively. Therefore, we conclude that the multilinear dimension reduction technique can help improve the prediction capabilities of our tensor-based regression model when the training sample size is not large enough. This is reasonable since multilinear dimensional reduction techniques help reduce the number of parameters to be estimated in the tensor-based models.

7.7 Case study: Degradation image streams from rotating machinery

In this section, we validate the effectiveness of our methodology using degradation image streams obtained from a rotating machinery. The experimental test bed, which was described in detail in [96], is designed to perform accelerated degradation tests on rolling element thrust bearings. The test bearings are run from a brand new state until failure. Vibration sensors are used to monitor the health of the rotating machinery. Failure is defined once the amplitude of defective vibration frequencies crosses a pre-specified threshold based on ISO standards for machine vibration. Meanwhile, infrared images that capture temperature signatures of the degraded component throughout the degradation test are acquired using an FLIR T300 infrared camera. Infrared images with 40×20 pixels are stored every 10 seconds. Four different experiments were run to failure. The resulting degradation-based image streams contained 375, 611, 827 and $1,478$ images, respectively.

Due to the high cost of running degradation experiments, additional degradation image streams were generated by resampling from the original image database obtained from the four experiments discussed earlier. In total 284 image data streams were generated. As an illustration, a sequence of images obtained at different (ordered) time periods are shown in Figure 7.5.



(a) t=1 (b) t=2 (c) t=3 (d) t=4 (e) t=5

(f) t=6 (g) t=7 (h) t=8 (i) t=9 (j) t=10

Figure 7.5: An illustration of one infrared degradation image stream.

7.7.1 Model selection

In this section, we discuss how to select an appropriate LLS tensor regression model for our dataset. This is achieved by applying different LLS tensor regression candidate models to a training set consisting of multiple image data streams. The model with the smallest BIC is selected as the best candidate model.

To account for the variability in the length of the image streams (as illustrated earlier in Section 7.4), we generate multiple subsamples based on different TTFs. Specifically, we sort the TTFs in ascending order such that $TTF_1 < TTF_2 < \cdots < TTF_n$, where $n \leq 284$ is the number of unique TTFs (or equivalent the number of subsamples). Next, we define subsample i as the systems whose TTFs are greater than or equal to TTF_i, for $i = 1, \ldots, n$. For example, subsample 1 includes all the 284 image streams, and subsample 2 includes all the image streams excluding the ones with the smallest TTF, and so forth. Third, each subsample is truncated by only keeping images observed on time domain $[0, TTF_i]$ epochs. By doing so, we ensure that all the image streams in a subsample have the same dimensionality. This is important when applying the LLS tensor regression model. After truncation,

the following steps are applied to select the best candidate regression model:

- *Step 1: Dimension reduction*. MPCA is applied to each subsample i (truncated image stream). The fraction-of-variance-explained, which is used to select the number of multilinear principal components (see [21] for details), is set to be 0.95. Using this criterion, a low-dimensional tensor is extracted from each image stream (or each system).

- *Step 2: Fitting LLS model*. The low-dimensional tensors extracted from Step 1 are regressed against TTFs using an LLS regression model. Similar to the Simulation study, we evaluate four types of distributions: *normal*, *lognormal*, *SEV* and *Weibull*. Tucker-based estimation method with heuristic rank selection is used for parameter estimation.

- *Step 3: Comparing BIC values.* BIC values are then computed for each of the four fitted models. The model with the smallest BIC is selected as the most appropriate one for the subsample.

- *Step 4: Distribution selection.* Steps 1, 2, and 3 are applied to all the subsamples. The distribution with the highest selected frequency is considered as the best candidate distribution.

After applying the aforementioned selection procedures to all the subsamples, we summarize the percentage of times each distribution was selected. Table 7.5 summarizes these results and shows that the *Weibull* distribution was selected on average 74.4% while the lognormal was selected 25.6% of the time. We expect to have some overlap in the models that have been selected because for specific parameter values, different distributions may exhibit reasonable fits for the same data sets. In our case, it is clear that the *Weibull* distribution dominates most of the selections and will therefore be considered as the suitable distribution for this data set.

Table 7.5: Distribution selection results.

LLS Distribution	Normal	Lognormal	SEV	Weibull
Selection (%)	0%	25.6%	0%	74.4%

7.7.2 Performance Evaluation

The Weibull tensor regression model is chosen for evaluating the accuracy of predicting lifetime. Similar to the simulation study in Section 7.6, we compare the performance of our methods with the "FPCA," "PCA," and "B-spline." Time-series degradation signals corresponding to the infrared images of the experimental test bed were obtained in a similar manner to what was discussed in Section 7.6. Figure 7.6 shows a sample of these transformed time-series signals.

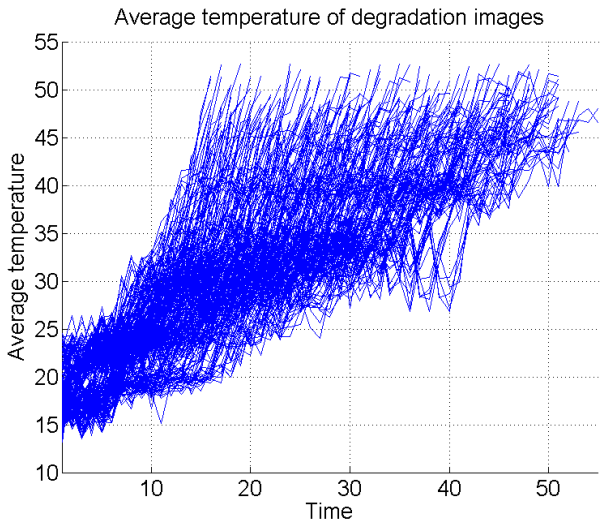
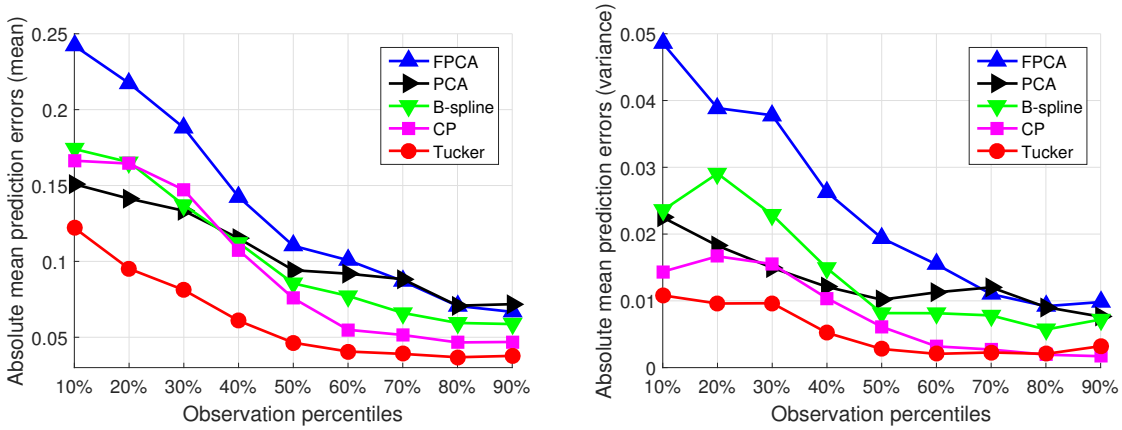


Figure 7.6: A sample of transformed time-series signals.

The accuracy and precision of the predictions made by the proposed model as well as the benchmarking models are evaluated using a leave-one-out cross-validation study. For each validation, 283 systems are used for training and the remaining one system is used for testing. The RULs of the test system are predicted at each time epoch. The time-varying regression framework presented in Section 7.4 is used to enable the integration of newly observed image data (from the test data). The prediction errors are computed

using Equation (2.25). We report the mean and variance of the absolute prediction errors in Figure 7.7 where 10% represents prediction errors evaluated at life percentiles in the interval of $(5\%, 15\%]$, 20% for the interval of $(15\%, 25\%]$, etc.



(a) mean (b) variance

Figure 7.7: The mean and variance of the absolute prediction errors.

Figure 7.7 indicates that all the five methodologies have smaller prediction errors at higher observation percentiles. This is because at higher observation percentiles more degradation-based image data has been observed, which provide more information about the underlying physical degradation process. This results in better prediction accuracy. Figure 7.7 also reveals that the proposed CP-based and Tucker-based regression models outperform "FPCA," "PCA," and "B-spline" in terms of mean and variance of the absolute prediction errors. For example, at the 50^{th} percentile, the mean (variance) of the absolute prediction errors for FPCA, PCA, B-spline, CP-based and Tucker-based models are $0.12(0.02)$, $0.09(0.009)$, $0.08(0.009)$, $0.07(0.006)$ and $0.05(0.003)$, respectively. A similar pattern can also be seen at most of the remaining prediction percentiles. This is consistent with the results in Section 7.6. Similarly, we believe this is because our tensor-based regression models are capable of (i) capturing the spatio-temporal correlation structure in each image stream, (ii) maximizing the captured stream-to-stream (system-to-system) variation in the low-dimensional projection space, and (iii) using TTF information to supervise the dimension reduction of the coefficient tensor. Again, comparing with our methods, none of

the benchmarks uses the TTF information for dimension reduction. In addition, "FPCA" results the loss of spatial information by averaging the pixel intensities. "PCA" is not able to maximize the captured system-to-system variation, breaks spatio-temporal correlation, and may lead to a large number of parameters. "B-spline" also fails to maximize the system-to-system variation. All these limitations compromise the prediction capabilities of the benchmarks.

Figure 7.7 also shows that Tucker-based regression performs better than CP-based. The mean and variance for the Tucker-based model are consistently lower than those of the CP-based regression model. This difference may be attributed to the fact that the Tucker-based model allows the tensor to have a different rank for each of the three orders (directions). This enhances the flexibility of the regression model. In contrast, the CP-based model requires the rank on each direction to be equal, which may have an impact on the model's flexibility.

7.8 Conclusions

Degradation tensors such as image streams and profiles often contain rich information about the physical degradation process and can be utilized for prognostics and predicting the RUL of functioning systems. However, the analysis of degradation tensors is challenging due to their high-dimensionality and complex spatial-temporal structure. In this chapter, we proposed a penalized (log)-location-scale regression model that can utilize high dimensional tensors to predict the RUL of systems. Our method first reduces the dimensionality of tensor covariates by projecting them onto a low-dimensional tensor subspace that preserves the useful information of the covariates. Next, the projected low-dimensional covariate tensors are regressed against TTFs via an LLS regression model. In order to further reduce the number of parameters, the coefficient tensor is decomposed by utilizing two tensor decompositions, CP and Tucker. The CP decomposition decomposes the coefficient tensor as a product of low-dimensional basis matrices, and Tucker decomposition expresses

it as a product of a low-dimensional core tensor and factor matrices. Instead of estimating the coefficient tensor, we only estimate its corresponding core tensors and factor/basis matrices. By doing so, the number of parameters to be estimated is dramatically reduced. Two numerical block relaxation algorithms with global convergence property were developed for the model estimation. The block relaxation algorithms iteratively estimate only one block of parameters (i.e., one factor/basis matrix or core tensor) at each time while keeping other blocks fixed until convergences.

We evaluated the performance of our proposed methodology through numerical studies. The results indicated that our methodology outperformed the benchmarks in terms of both prediction accuracy and precision. In addition, the results showed that the multilinear dimension reduction technique could help improve the prediction accuracy and precision, especially when the training sample size is small. We also validated the effectiveness of our proposed tensor regression model using a case study on degradation modeling of bearings in a rotating machinery. The results indicated that both CP-based and Tucker-based models outperformed the benchmarks in terms of prediction accuracy as well as precision at almost all the life percentiles. The results also indicated that Tucker-based model achieved better prediction accuracy than the CP-based model. This is reasonable since Tucker-based model is more flexible as it allows different modes to have different ranks, while the CP-based model requires all the modes have the same rank. The model developed in this chapter only works on a single computer. Development of a tensor-based prognostics model that can run on a distributed computing system is an important topic for future research.

CHAPTER 8
CONCLUSIONS AND FUTURE RESEARCH

8.1 Conclusions

This thesis has presented new predictive analytics methodologies that extract information from massive and complex-structured high-dimensional signals with the goal of predicting (in real-time) the future state-of-health of complex engineering systems. The main research results and new contributions of this dissertation are summarized as follows.

(1) A new methodology was developed for systematically selecting informative sensors, fusing multi-stream signals, and predicting residual useful lifetimes. This is achieved by building a penalized LLS functional regression model, which integrates LLS functional regression and group nonnegative garrote. The LLS functional model regresses degradation trajectories against TTFs, and the coefficient functions are penalized using group nonnegative garrote. To address the model estimation challenge, FPCA is employed to transform the penalized LLS functional regression to penalized LLS regression. The transformed model is then solved using penalized maximum likelihood estimation and informative sensors are selected. The informative sensors are then fused utilizing multivariate FPCA to predict remaining operational lifetimes. Using multivariate sensor data from an aircraft turbofan engine consisting of 21 sensors, we were able to achieve higher prediction accuracy using 4 sensors selected by our approach relative to the original 21 sensors.

(2) A new scalable prognostic model was proposed for large-scale condition monitoring applications. The proposed methodology focuses on computational scalability of the functional data analysis-based prognostic framework, which utilizes multivariate FPCA to fuse the multi-stream high-dimensional degradation signals and then uses the resulting features to predict the TTF. Classic multivariate FPCA typically involves some form of decom-

position or factorization of a matrix (or covariance matrix) constructed from multi-stream signals. Such decomposition/factorization is often computationally infeasible since the matrix is usually extremely large given the large size and high dimensionality of the data. This thesis addresses this challenge by integrating randomized low-rank matrix approximation with multivariate FPCA computations. Randomized low-rank matrix approximation computes the leading singular values and vectors of the signal matrix via randomized sampling. This is achieved by first computing an approximation to the range (also known as column space) of a matrix via randomized sampling. The signal matrix is then projected to the approximated range and a factorization of the resulting low-rank matrix is computed. Using a numerical study, we showed that the computational time for predicting remaining lifetime distribution of 100 units with 1,000 sensors per unit using best-in class models required 24 minutes compared to 10 seconds using the proposed approach (without loss in accuracy).

(3) A novel adaptive functional regression-based prognostics model was developed for applications in which degradation signals have different forms of missing data, i.e., sparse or fragmented data. The methodology was based on using FPCA to identify a general nonparametric trend for degradation signals pertaining to a population of similar components. An adaptive functional regression model was then used to model the relationship between the FPC-scores and the time-to-failure of the components. Real-time signals observed from validation components (assumed to be operating in the field) were incorporated into the model and used to update the predicted time-to-failure of each fielded component based on their unique degradation characteristics. The model was validated using two sets of degradation data, crack growth and bearing vibration data. The performance of the model was benchmarked against other nonparametric and parametric models. The investigation was performed for complete, sparse, and fragmented signal scenarios. Results indicated that the performance of our proposed model was more robust compared and provided relatively failure predictability in comparison to the other benchmarks used in the study. This was particularly true to for sparse and fragmented degradation signals. In the case of complete

signals that had no missing data, our model performance at least as good as the benchmark parametric model.

(4) A new robust prognostic model was proposed for applications with large amount of incomplete multi-stream signals. We proposed two algorithms that use matrix completion to address the missing data challenge for multi-sensor applications. The first algorithm, the subspace detection method, uses matrix completion techniques to compute a set of basis that spans the column space of the original signal matrix. The basis are then utilized by a novel-developed algorithm to extract signal features. The second algorithm, the signal recovery method, involves two steps, conventional matrix completion followed by feature extraction. Matrix completion techniques are employed to recover the missing degradation data of each sensor individually. Recovered signals are then utilized to extract signal features via a newly-developed incremental SVD algorithm, which significantly helps reduce the computational complexity and memory requirement. The proposed methodologies were evaluated through an extensive numerical study and real-world data. The results demonstrated that the proposed approaches are robust to significant levels of missing data and can maintain reasonable prediction accuracy even if the signals are highly incomplete.

(5) A novel prognostic model for multi-sensor applications with highly-incomplete degradation signals and censored historical failure times was developed. The methodology builds an optimization problem combining a feature extraction term and a regression term. The feature extraction term extracts low-dimensional features of multi-stream degradation signals using their incomplete observations, and the regression term regresses the features against the censored TTFs. By simultaneously optimizing the two terms, the TTFs are used to supervise the feature extraction process, and thus the extracted features are guaranteed to be most informative for TTF prediction. To solve the optimization problem, we developed a Block Prox-Linear Coordinate Descent algorithm and theoretically proved its global convergence property. A simulated dataset and a multi-stream degradation data from aircraft turbofan engines were used to evaluate the performance of our proposed methodology. The

results indicated that our proposed methodology achieved high prediction accuracy even if the degradation signals are highly incomplete and the historical failure times present a significant level of censoring. In addition, the results also illustrated that our model consistently outperformed the unsupervised dimension reduction-based benchmarks in terms of prediction errors, at all levels of data incompleteness and failure time censoring, which confirmed the importance of using failure times to supervise the feature extraction (or dimension reduction) process in prognostic modeling.

(6) A new prognostic model was proposed for industrial applications involving image data. The methodology integrates tensor linear algebra with traditional LLS regression widely used in reliability and prognostics. To address the high dimensionality challenge, the degradation image streams are first projected to a low-dimensional tensor subspace that is able to preserve their information. Next, the projected image tensors are regressed against TTFs via penalized LLS tensor regression. The coefficient tensor is then decomposed using CP and Tucker decompositions, which enables parameter estimation in a high-dimensional setting. Two optimization algorithms with a global convergence property were developed for model estimation. The effectiveness of the proposed models was validated using two simulated datasets and infrared degradation image streams from a rotating machinery.

8.2 Future research

Predictive analytics using high-dimensional signals is an important yet challenging research problem. In this dissertation, we have made some initial efforts in this area. In the future, one important and interesting research opportunity is to extend the current predictive modeling framework to other practical scenarios that include: (1) settings involving multiple failure modes, which are common among most industrial assets, and (2) applications where assets function under different operational modes and/or loading profiles. Different failure modes stem from different physical degradation processes that can be mutually exclusive (a simple case) or interdependent (a more complex case). The sensors that measure these

failure modes can be distinct for each mode (a simple case) or overlap with other modes (a more complex scenario). Similarly, varying operating conditions have varying effects on degradation rates, e.g., more harsh environments tend to accelerate the degradation process. These problems require fundamental research that aims at advancing conventional statistical and stochastic methodologies to enable modeling of such complex settings that have exciting research challenges yet remain strongly relevant to real-world applications.

The above problems generate their own unique computational and scalability challenges. Many analytic models require re-estimating several components of the existing model from scratch every time new data is observed. This makes the computation even more challenging in settings involving multi-failure modes and time-varying operating conditions. Rather than performing major re-computations, my plan is to focus on developing sequential algorithms that maximize the utilization of previous model estimation during the sensor-based updatingcurrently one of the main research challenges in online predictive analytics.

Appendices

APPENDIX A

SUPPLEMENTARY MATERIALS OF CHAPTER 2

The location parameter in criterion (2.8) is expressed as follows:

$$\pi(s_{i,p}(t)) = \tilde{\alpha}_0 + \sum_{p=1}^{P} \int_0^T \tilde{\alpha}_p(t) s_{i,p}(t) dt \tag{A.1}$$

Recall that $\tilde{\alpha}_p(t) = \hat{\alpha}_p(t) d_p(t)$, $\hat{\alpha}_p(t) = \sum_{k=1}^{\infty} \beta_{k,p} \phi_{k,p}(t)$ and $s_{i,p}(t) = \mu_p(t) + \sum_{k=1}^{\infty} \xi_{i,k,p} \phi_{k,p}(t) + \epsilon_{i,p}(t)$, we have

$$\begin{aligned}
\pi(s_{i,p}(t)) &= \tilde{\alpha}_0 + \sum_{p=1}^{P} \int_0^T \tilde{\alpha}_p(t) \left\{ \mu_p(t) + \sum_{k=1}^{\infty} \xi_{i,k,p} \phi_{k,p}(t) + \epsilon_{i,p}(t) \right\} \\
&= \tilde{\alpha}_0 + \sum_{p=1}^{P} \int_0^T \tilde{\alpha}_p(t) \mu_p(t) dt + \sum_{p=1}^{P} \int_0^T \left\{ \hat{\alpha}_p(t) d_p \left(\sum_{k=1}^{\infty} \xi_{i,k,p} \phi_{k,p}(t) \right) \right\} dt \\
&+ \sum_{p=1}^{P} \int_0^T \tilde{\alpha}_p(t) \epsilon_{i,p}(t) dt \\
&= \beta_0 + \sum_{p=1}^{P} \int_0^T \left\{ d_p \sum_{k=1}^{\infty} \beta_{k,p} \phi_{k,p}(t) \right\} \left\{ \sum_{k=1}^{\infty} \xi_{i,k,p} \phi_{k,p}(t) \right\} dt + \varepsilon_i \\
&= \beta_0 + \sum_{p=1}^{P} d_p \sum_{k=1}^{\infty} \beta_{k,p} \xi_{i,k,p} + \varepsilon_i \\
&= \beta_0 + \sum_{p=1}^{P} d_p \sum_{k=1}^{K_p} \beta_{k,p} \xi_{i,k,p},
\end{aligned} \tag{A.2}$$

where $\beta_0 = \tilde{\alpha}_0 + \sum_{p=1}^{P} \int_0^T \tilde{\alpha}_p(t) \mu_p(t) dt$ is the intercept.

APPENDIX B

SUPPLEMENTARY MATERIALS OF CHAPTER 3

Recall that $\pi_i(s_{i,p}(t)) = \alpha_0 + \int_0^T \boldsymbol{\alpha}(t)^\top \boldsymbol{s}_i(t)dt$, $\boldsymbol{\alpha}(t) = \sum_{k=1}^{\infty} \beta_k \boldsymbol{\psi}_k(t)$ and $\boldsymbol{s}_i(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{\infty} \zeta_{i,k}\boldsymbol{\psi}_k(t)$,

$$\begin{aligned}
\pi_i(s_{i,p}(t)) &= \alpha_0 + \int_0^T \boldsymbol{\alpha}(t)^\top \boldsymbol{s}_i(t)dt \\
&= \alpha_0 + \int_0^T \boldsymbol{\alpha}(t)^\top \boldsymbol{\mu}(t)dt + \int_0^T \left\{\sum_{k=1}^{\infty} \beta_k \boldsymbol{\psi}_k(t)^\top\right\}\left\{\sum_{k=1}^{\infty} \zeta_{i,k}\boldsymbol{\psi}_k(t)\right\} dt \\
&= \beta_0 + \sum_{k=1}^{\infty} \beta_k \zeta_{i,k} \\
&\approx \beta_0 + \sum_{k=1}^{K} \beta_k \zeta_{i,k},
\end{aligned}$$

where $\beta_0 = \alpha_0 + \int_0^T \boldsymbol{\alpha}(t)^\top \boldsymbol{\mu}(t)dt$.

APPENDIX C

SUPPLEMENTARY MATERIALS OF CHAPTER 4

Due to the missing signal observations, the model expressed in equation (4.5) is estimated using the "pooled" historical degradation data. This allows us to borrow information across different components, which improves the estimation process. First, we denote the degradation signal of component i, as $S_i(t_{ij})$, where $j = 1, \ldots, m_i$, and m_i is the number of observation time points of signal i. (Recall, $i = 1, \ldots, n$, and n is the number of signals). $\{t_{ij}\}_{j=1,\ldots,m_i}$ are sparsely observed times points on the time domain $[0, M]$ for signal i.Using these notation, we arrive at the following form:

$$\begin{aligned} S_i(t_{ij}) &= \mu(t_{ij}) + X_i(t_{ij}) + \epsilon_i(t_{ij}) \\ &= \mu(t_{ij}) + \sum_{k=1}^{K} \xi_{ik}\phi_k(t_{ij}) + \epsilon_i(t_{ij}). \end{aligned} \tag{C.1}$$

Next, we illustrate how to estimate the mean and covariance functions, and the FPC-scores.

C.1 Estimating the Mean Function

The mean function $\mu(t)$ is estimated using a smoothing technique, specifically local regression [45]. In local regression, a smoothing window with a given bandwidth is defined around a point in the domain of the mean function. A smooth function is then approximated in that neighborhood using the available signal observations. We use a local linear approximation of the mean function, i.e, $\mu(t_{ij}) \approx c_0 + c_1(t_{ij} - t)$. The estimates of the

linear coefficients, $\hat{c}_0$ and $\hat{c}_1$ are chosen to minimize the following function:

$$\min_{c_0,c_1} \sum_{i=1}^{n} \sum_{j=1}^{m_i} W(\frac{t_{ij}-t}{d_\mu})\{S_i(t_{ij}) - c_0 - c_1(t-t_{ij})\}^2, \tag{C.2}$$

where t is the objective fitting time point, t_{ij} are the observed time points within the smoothing window, d_μ is the bandwidth of the smoothing window, which is selected by using the one-curve-leave-out cross-validation method (see [47] for additional details), and $W(\cdot)$ is a Gaussian kernel function that assigns more weight to those observations close to point t. $\mu(t)$ is estimated as follows; $\hat{\mu}(t) = \hat{c}_0(t)$.

C.2 Estimating the Covariance Function

The covariance function $C(t,t')$ is estimated using the demeaned signal data $S_i(t_{ij}) - \hat{\mu}(t_{ij})$, where $\hat{\mu}(t)$ is the estimated mean function obtained from the previous step. The raw covariance surface is denoted by $G_i(t_{ij},t_{ik}) = Cov(S_i(t_{ij}) - \hat{\mu}(t_{ij}), S_i(t_{ik}) - \hat{\mu}(t_{ik}))$. Recall that $C_i(t_{ij},t_{ik}) = Cov(X_i(t_{ij}), X_i(t_{ik}))$, thus we have $G_i(t_{ij},t_{ik}) = C_i(t_{ij},t_{ik}) + \sigma^2\delta_{t_{ij}t_{ik}}$, where $\delta_{t_{ij}t_{ik}} = 1$, if $t_{ij} = t_{jk}$, and 0 otherwise. In order to estimate $C_i(t_{ij},t_{ik})$, we only consider the off-diagonal elements of the raw covariance surface $G_i(t_{ij},t_{ik})$ (since the diagonal elements contain an additional element, the error variance). In other words, only $G_i(t_{ij},t_{ik}), t_{ij} \neq t_{ik}$ are used to estimate $C_i(t_{ij},t_{ik})$. We also use local regression to estimate the covariance function as follows [45]:

$$\min_{c_0,c_1,c_2} \sum_{i=1}^{n} \sum_{1\leq j\neq k\leq m_i} W(\frac{t_{ij}-t}{d_G}, \frac{t_{ik}-t'}{d_G})\{G_i(t_{ij},t_{ik}) - c_0 - c_1(t-t_{ij}) - c_2(t'-t_{ik})\}^2, \tag{C.3}$$

where d_G is the smoothing window bandwidth, which is also selected by using one-curve-leave-out cross-validation, and $W(\cdot)$ is a bivariate Gaussian kernel function.

$C(t,t')$ is estimated as $\hat{C}(t,t') = \hat{c}_0(t,t')$. Recall that $C(t,t') = \sum_{k=1}^{\infty} \lambda_k \phi_k(t)\phi_k(t')$ where the eigenfunctions $\phi_k(t)$ and the eigenvalues λ_k can now be estimated by solving the

following eigen equations:

$$\int_{[0,M]} \hat{C}(t,t')\hat{\phi}_k(t)dt = \hat{\lambda}_k\hat{\phi}_k(t'), \tag{C.4}$$

where $\int_{[0,M]} \hat{\phi}_k(t) \times \hat{\phi}_m(t)dt = 1$, if $m = k$, and 0 otherwise. Equation (C.4) is solved by discretizing the estimated covariance surface $\hat{C}(t,t')$ (details can be found in [47]).

C.3 Estimating the Error Term

To estimate the error terms, we let $\hat{D}(t)$ represent a smoothed function estimated using the diagonal elements of the raw covariance surface i.e. $C(t,t') + \sigma^2$. Next, we define $\tilde{C}(t)$ as a smoothed function based on the covariance surface, $C(t,t')$. In other words, $\hat{D}(t)$ is calculated based on the "raw" diagonal elements, and hence it contains the variance of the error term, whereas $\tilde{C}(t)$ is estimated based on covariance matrix with the "smoothed" diagonal elements, and hence so it does not contain the variance of the error term. Using these two quantities, we can estimate σ^2 using the following expression:

$$\hat{\sigma}^2 = \frac{1}{|M|}\int_{[0,M]} \{\hat{D}(t) - \tilde{C}(t)\}dt, \tag{C.5}$$

where $|M|$ denotes the length of $[0, M]$.

$\hat{D}(t)$ is estimated using local linear smoothers similar to those presented in Equation (C.2) with $G_i(t_{ij}, t_{ij})$ as input. Local quadratic smoothers are used to estimate $\tilde{C}(t)$ since they tend to capture the shape of the surface better. Consequently, local quadratic smoothers are used to estimate $\tilde{C}(t)$ by rotating the coordinates by 45 degrees as proposed in [**fpca**] (since the covariance surface $C(t,t')$ is maximal along the diagonal as noted by [129]). By rotating the axes by 45 degrees clockwise, we have the following;

$$\begin{pmatrix} t^r_{ij} \\ t^r_{ik} \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \begin{pmatrix} t_{ij} \\ t_{ik} \end{pmatrix}$$

where (t_{ij}, t_{ik}) denote the points in the original axes, and (t^r_{ij}, t^r_{ik}) represent the corresponding points on the rotated axes.

The rotated $C(t, t')$ is estimated using the quadratic local smoothers as follows:

$$\min_{c_0,c_1,c_2} \sum_{i=1}^{n} \sum_{1\le j\neq k\le m_i} W(\frac{t^r_{ij}-t}{d_G}, \frac{t^r_{ik}-t'}{d_G})\{G_i(t^r_{ij}, t^r_{ik}) - c_0 - c_1(t-t^r_{ij}) - c_2(t'-t^r_{ik})^2\}^2, \quad \text{(C.6)}$$

where d_G is the smoothing bandwidth; and $W(\cdot)$ is the bivariate Gaussian kernel function. If we let $\hat{C}^r(t) = \hat{c}_0(t, t')$ be the rotated estimate of $C(t, t')$, then $\tilde{C}(t) = \hat{C}^r(0, t/\sqrt{2})$, and hence, σ^2 can be estimated using Equation C.5.

C.4 Estimating the FPC-scores

Typically, with complete degradation signals the FPC-scores are given as $\xi_{ik} = \int_{[0,M]}(X_i(t) - \mu(t))\phi_k(t)dt$ and can be estimated by numerical integration using $\hat{\xi}_{ik} = \sum_{j=1}^{m_i}(S_i(t_{ij}) - \hat{\mu}(t_{ij}))\hat{\phi}_k(t_{ij})(t_{ij} - t_{i(j-1)})$, where $t_{i0} = 0$. However, when dealing with sparse and fragmented signals, numerical integration may not be suitable, especially when the degradation signals are too sparse. Consequently, we use a method proposed by [**fpca**] and is known as Principal Analysis by Conditional Expectation (PACE). Asymptotic results reported by the authors demonstrate that the PACE method is well-suited for estimating FPC-scores when repeated measurements are irregularly spaced and the number of observations per subject is limited. To illustrate this method, we begin by defining the following vectors; $\tilde{\mathbf{X}}_i = (X_i(t_{i1}), ..., X_i(t_{im_i}))^T$, $\widetilde{\mathbf{S}}_i = (S_i(t_{i1}), ..., S_i(t_{im_i}))^T$, $\hat{\boldsymbol{\mu}}_i = (\hat{\mu}(t_{i1}), ..., \hat{\mu}(t_{im_i}))^T$, and $\hat{\boldsymbol{\phi}}_{ik} = (\hat{\phi}_k(t_{i1}), ..., \hat{\phi}_k(t_{im_i}))^T$. Based on the assumption that ξ_{ik} and ϵ_i are jointly Gaussian, the FPC-scores of the ith signal can be estimated using the following conditional expectation:

$$\hat{\xi}_{ik} = \hat{E}[\xi_{ik}|\tilde{\mathbf{S}}_i] = \hat{\lambda}_k \hat{\boldsymbol{\phi}}^T_{ik} \hat{\boldsymbol{\Sigma}}^{-1}_{\mathbf{S}_i}(\tilde{\mathbf{S}}_i - \hat{\boldsymbol{\mu}}_i), \quad \text{(C.7)}$$

where $\hat{\mathbf{\Sigma}}_{\mathbf{S}_i} = cov(\tilde{\mathbf{S}}_i, \tilde{\mathbf{S}}_i) = cov(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_i) + \hat{\sigma}^2 \mathbf{I}_{m_i}$. $\hat{\mathbf{\Sigma}}_{\mathbf{S}_i}$ is a $m_i \times m_i$ matrix and its $(j,k)th$ element is $\hat{\mathbf{\Sigma}}_{\mathbf{S}_i}(j,k) = \hat{C}(t_{ij}, t_{ik}) + \hat{\sigma}^2 \delta_{t_{ij}t_{ik}}$, where $\delta_{t_{ij}t_{ik}} = 1$, if $t_{ij} = t_{jk}$ and 0 otherwise.

APPENDIX D

SUPPLEMENTARY MATERIALS OF CHAPTER 5

Since $\boldsymbol{Q}$ is an orthonormal basis of $\boldsymbol{S}$, we have $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{S}$ and $\tilde{\boldsymbol{S}} = \boldsymbol{Q}\boldsymbol{Q}^\top \tilde{\boldsymbol{S}}$. Since $\boldsymbol{W} = \boldsymbol{Q}^\top \boldsymbol{S}$, then $\overline{\boldsymbol{W}} = \boldsymbol{Q}^\top \overline{\boldsymbol{S}}$. Thus, we have $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{S} = \boldsymbol{Q}\boldsymbol{W} = \boldsymbol{Q}(\tilde{\boldsymbol{W}} + \overline{\boldsymbol{W}}) = \boldsymbol{Q}\tilde{\boldsymbol{W}} + \boldsymbol{Q}\overline{\boldsymbol{W}} = \boldsymbol{Q}\tilde{\boldsymbol{W}} + \boldsymbol{Q}\boldsymbol{Q}^\top \overline{\boldsymbol{S}} \Longrightarrow \boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{S} = \boldsymbol{Q}\tilde{\boldsymbol{W}} + \boldsymbol{Q}\boldsymbol{Q}^\top \overline{\boldsymbol{S}} \Longrightarrow \boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{S} - \boldsymbol{Q}\boldsymbol{Q}^\top \overline{\boldsymbol{S}} = \boldsymbol{Q}\tilde{\boldsymbol{W}} \Longrightarrow \boldsymbol{Q}\boldsymbol{Q}^\top (\boldsymbol{S} - \overline{\boldsymbol{S}}) = \boldsymbol{Q}\tilde{\boldsymbol{W}} \Longrightarrow \boldsymbol{Q}\boldsymbol{Q}^\top \tilde{\boldsymbol{S}} = \boldsymbol{Q}\tilde{\boldsymbol{W}} \Longrightarrow \tilde{\boldsymbol{S}} = \boldsymbol{Q}\tilde{\boldsymbol{W}}$. In addition, since $\tilde{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, we can conclude $\tilde{\boldsymbol{S}} = \boldsymbol{Q}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\top$, where $\hat{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{U}, \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}, \hat{\boldsymbol{V}} = \boldsymbol{V}$.

APPENDIX E

SUPPLEMENTARY MATERIALS OF CHAPTER 6

In this section, we show the proofs of the lemmas, propositions and theorems used in this chapter.

E.1 Preliminaries

Lemma E.1.1 *The negative log-likelihood functions of (log)-location-scale (LLS) distributions, denoted by $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}})$, have the following property:*

$$\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) \geq -n\log(\tilde{\sigma}) + \|\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}\|_1 \tag{E.1}$$

where $\|\cdot\|_1$ is the ℓ_1 norm.

Proof 2 *For an SEV/Weibull distribution, it is known that $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = -n\log\tilde{\sigma} - \sum_{i=1}^n(\tilde{\sigma}y_i - \tilde{\beta}_0 - \boldsymbol{u}_i\tilde{\boldsymbol{\beta}}) + \sum_{i=1}^n\exp(\tilde{\sigma}y_i - \tilde{\beta}_0 - \boldsymbol{u}_i\tilde{\boldsymbol{\beta}})$. Let $\tilde{\omega}_i = \tilde{\sigma}y_i - \tilde{\beta}_0 - \boldsymbol{u}_i\tilde{\boldsymbol{\beta}}$, we have the following:*

$$\begin{aligned}
&\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) \\
&= -n\log\tilde{\sigma} - \sum_i\tilde{\omega}_i + \sum_i\exp(\tilde{\omega}_i) \\
&= -n\log\tilde{\sigma} - \sum_{i,\tilde{\omega}_i\geq 0}\tilde{\omega}_i + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\exp(\tilde{\omega}_i) + \sum_{i,\tilde{\omega}_i<0}\exp(\tilde{\omega}_i) \\
&\geq -n\log\tilde{\sigma} - \sum_{i,\tilde{\omega}_i\geq 0}\tilde{\omega}_i + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\exp(\tilde{\omega}_i) \\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}(\exp(\tilde{\omega}_i) - \tilde{\omega}_i)
\end{aligned}$$

$$
\begin{aligned}
&\geq -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\left(1+\tilde{\omega}_i+\frac{\tilde{\omega}_i^2}{2}-\tilde{\omega}_i\right) \text{[\textit{Taylor expansion}]}\\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\left(\frac{(\tilde{\omega}_i-1)^2}{2}+\tilde{\omega}_i+\frac{1}{2}\right)\\
&> -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\tilde{\omega}_i\\
&= -n\log\tilde{\sigma} + \sum_{i}|\tilde{\omega}_i|\\
&= -n\log\tilde{\sigma} + \|\tilde{\sigma}\boldsymbol{y}-\mathbf{1}_n\tilde{\beta}_0-\boldsymbol{U}\tilde{\boldsymbol{\beta}}\|_1
\end{aligned}
$$

For a logistics/loglogistics distribution, it is known that $\ell(\tilde{\sigma}\boldsymbol{y}-\mathbf{1}_n\tilde{\beta}_0-\boldsymbol{U}\tilde{\boldsymbol{\beta}}) = -n\log\tilde{\sigma} - \sum_{i=1}^{n}(\tilde{\sigma}y_i-\tilde{\beta}_0-\boldsymbol{u}_i\tilde{\boldsymbol{\beta}}) + 2\sum_{i=1}^{n}\log(1+\exp(\tilde{\sigma}y_i-\tilde{\beta}_0-\boldsymbol{u}_i\tilde{\boldsymbol{\beta}}))$*. We have the following:*

$$
\begin{aligned}
&\ell(\tilde{\sigma}\boldsymbol{y}-\mathbf{1}_n\tilde{\beta}_0-\boldsymbol{U}\tilde{\boldsymbol{\beta}})\\
&= -n\log\tilde{\sigma} - \sum_{i}\tilde{\omega}_i + 2\sum_{i}\log(1+\exp(\tilde{\omega}_i))\\
&= -n\log\tilde{\sigma} - \sum_{i,\tilde{\omega}_i\geq 0}\tilde{\omega}_i + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + 2\sum_{i,\tilde{\omega}_i\geq 0}\log(1+\exp(\tilde{\omega}_i)) + 2\sum_{i,\tilde{\omega}_i<0}\log(1+\exp(\tilde{\omega}_i))\\
&> -n\log\tilde{\sigma} - \sum_{i,\tilde{\omega}_i\geq 0}\tilde{\omega}_i + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + 2\sum_{i,\tilde{\omega}_i\geq 0}\log(1+\exp(\tilde{\omega}_i))\\
&[\because (1+\exp(\tilde{\omega}_i)) > 1, \therefore \log(1+\exp(\tilde{\omega}_i)) > 0]\\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}(2\log(1+\exp(\tilde{\omega}_i)) - \tilde{\omega}_i)\\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}(2\log(1+\exp(\tilde{\omega}_i)) - \log\exp(\tilde{\omega}_i))\\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}(2\log(1+\exp(\tilde{\omega}_i)) - \log\exp(\tilde{\omega}_i))\\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\left(\log\frac{(1+\exp(\tilde{\omega}_i))^2}{\exp(\tilde{\omega}_i)}\right)\\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0}|\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0}\left(\log\left(\exp(\tilde{\omega}_i)+2+\frac{1}{\exp(\tilde{\omega}_i)}\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
&> -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0} |\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0} (\log(\exp(\tilde{\omega}_i)) \\
&= -n\log\tilde{\sigma} + \sum_{i,\tilde{\omega}_i<0} |\tilde{\omega}_i| + \sum_{i,\tilde{\omega}_i\geq 0} \tilde{\omega}_i \\
&= -n\log\tilde{\sigma} + \sum_{i} |\tilde{\omega}_i| \\
&= -n\log\tilde{\sigma} + \|\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}\|_1
\end{aligned}
$$

For a normal/lognormal distribution, it is known that $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = \frac{n}{2}\log 2\pi - n\log\tilde{\sigma} + \frac{1}{2}\sum_{i=1}^{n}(\tilde{\sigma}y_i - \tilde{\beta}_0 - \boldsymbol{u}_i\tilde{\boldsymbol{\beta}})^2$*.We have the following:*

$$
\begin{aligned}
&\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) \\
&= \frac{n}{2}\log 2\pi - n\log\tilde{\sigma} + \frac{1}{2}\sum_{i=1}^{n}\tilde{\omega}_i^2 \\
&> -n\log\tilde{\sigma} + \frac{1}{2}\sum_{i=1}^{n}(\tilde{\omega}_i^2 + 1) \;\; [\because \frac{n}{2}\log 2\pi > \frac{n}{2}] \\
&= -n\log\tilde{\sigma} + \frac{1}{2}\sum_{i=1}^{n}\left((|\tilde{\omega}_i| - 1)^2 + 2|\tilde{\omega}|_i\right) \\
&\geq -n\log\tilde{\sigma} + \sum_{i=1}^{n}|\tilde{\omega}_i| \\
&= -n\log\tilde{\sigma} + \|\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}\|_1
\end{aligned}
$$

E.2 Proof of Lemma 6.3.1

Without loss of generality, we let $\boldsymbol{U}^0 = \mathbf{0}$, $\boldsymbol{V}^0 = \mathbf{0}$, $y_i^0 = c_i$ if $i \notin O$ and $y_i^0 = t_i$ if $i \in O$, $\tilde{\beta}_0^0 = 0$, $\tilde{\boldsymbol{\beta}}^0 = \mathbf{0}$, $\tilde{\sigma}^0 = 1$, then we have $\tilde{\mathcal{F}}(\boldsymbol{\theta}^0) = C$, where $\boldsymbol{\theta}^0 = \{\boldsymbol{U}^0, \boldsymbol{V}^0, \boldsymbol{y}^0, \tilde{\beta}_0^0, \tilde{\boldsymbol{\beta}}^0, \tilde{\sigma}^0\}$ and C is a constant. For any $\boldsymbol{\theta}^k$, if $\tilde{\mathcal{F}}(\boldsymbol{\theta}^k) \leq \tilde{\mathcal{F}}(\boldsymbol{\theta}^0)$ holds, then we have the following

inequality:

$$\|\mathcal{P}_\Omega(\boldsymbol{S}-\boldsymbol{U}^k\boldsymbol{V}^k)\|_F^2+\lambda_1(\|\boldsymbol{U}^k\|_F^2+\|\boldsymbol{V}^k\|_F^2)+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)+\lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\leq C \tag{E.2}$$

E.2.1 $\tilde{\sigma}^k$

We first prove that $\tilde{\sigma}^k$ is bounded by using contradiction. From (E.2), we have the following:

$$\begin{aligned} & w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)\leq C \\ \Longrightarrow\ & w(-n\log(\tilde{\sigma}^k)+\|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1)\leq C \ \text{ (Lemma E.1.1)} \\ \Longrightarrow\ & -wn\log(\tilde{\sigma}^k)\leq C \end{aligned}$$

If $\tilde{\sigma}^k\to 0$, then $-wn\log(\tilde{\sigma}^k)\to\infty$, contradiction! Therefore, we conclude $\tilde{\sigma}^k\neq 0$. Next, we prove that $\tilde{\sigma}^k\nrightarrow\infty$. From (E.2), we have the following:

$$\begin{aligned} & w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)+\lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\leq C \\ \Longrightarrow\ & w(-n\log(\tilde{\sigma}^k)+\|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1)+\lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\leq C \ \text{ (Lemma E.1.1)} \end{aligned}$$

Recall in criterion (6.5), we applied the following transformation: $\tilde{\sigma}=1/\sigma,\tilde{\beta}_0=\beta_0/\sigma,\tilde{\boldsymbol{\beta}}=\boldsymbol{\beta}/\sigma$. Here, for the convenience of proof, we use term $\boldsymbol{\beta},\beta_0,\boldsymbol{\beta}$. As a result, we have the following:

$$wn\log(\sigma^k)+w\sum_{i=1}^n\left|\frac{y_i^k-\beta_0^k-\boldsymbol{u}_i^k\boldsymbol{\beta}^k}{\sigma^k}\right|+\lambda_2\sum_{j=1}^p\left|\frac{\beta_j^k}{\sigma^k}\right|\leq C \tag{E.3}$$

Lemma E.2.1 *For* (E.3)*, if* $\sigma^k=0$*, then we have following two equations:*

$$y_i^k=\beta_0^k+\boldsymbol{u}_i^k\boldsymbol{\beta}^k,\forall i \tag{E.4}$$

$$\beta_j^k=0,\forall j \tag{E.5}$$

Proof 3 *We first prove equation* (E.4)*. Assume there exists an* $\tilde{i}$ *such that* $y_{\tilde{i}}^k\neq\beta_0^k+\boldsymbol{u}_{\tilde{i}}^k\boldsymbol{\beta}^k$*, then from* (E.3)*, we have*

$$
\begin{aligned}
& wn\log(\sigma^k) + w\textstyle\sum_{i=1}^{n}\left|\frac{y_i^k-\beta_0^k-\boldsymbol{u}_i^k\boldsymbol{\beta}^k}{\sigma^k}\right| + \lambda_2\textstyle\sum_{j=1}^{p}\left|\frac{\beta_j^k}{\sigma^k}\right| \leq C \\
\Longrightarrow \quad & wn\log(\sigma^k) + w\textstyle\sum_{i=1}^{n}\left|\frac{y_i^k-\beta_0^k-\boldsymbol{u}_i^k\boldsymbol{\beta}^k}{\sigma^k}\right| \leq C \\
\Longrightarrow \quad & wn\log(\sigma^k) + w\left|\frac{y_{\tilde{i}}^k-\beta_0^k-\boldsymbol{u}_{\tilde{i}}^k\boldsymbol{\beta}^k}{\sigma^k}\right| \leq C
\end{aligned}
$$

Based on l'Hôpital's Rule $\lim_{x\to 0}\frac{\log(x)}{1/x} = \lim_{x\to 0}\frac{1/x}{-1/x^2} = 0$. *Therefore,* $\lim_{x\to 0}\{\log(x)+1/x\} \to \infty$. *As a result, we have the following:*

$$
wn\log(\sigma^k) + w\left|\frac{y_{\tilde{i}}^k-\beta_0^k-\boldsymbol{u}_{\tilde{i}}^k\boldsymbol{\beta}^k}{\sigma^k}\right| \leq C \Longrightarrow \quad \infty \leq C
$$

Contradiction! Thus, (E.4) *holds. Next, we prove that* (E.5) *also holds. Assume there exists an* $\tilde{j}$ *such that* $\beta_{\tilde{j}}^k \neq 0$*, then from* (E.3)*, we have*

$$
\begin{aligned}
& wn\log(\sigma^k) + w\textstyle\sum_{i=1}^{n}\left|\frac{y_i^k-\beta_0^k-\boldsymbol{u}_i^k\boldsymbol{\beta}^k}{\sigma^k}\right| + \lambda_2\textstyle\sum_{j=1}^{p}\left|\frac{\beta_j^k}{\sigma^k}\right| \leq C \\
\Longrightarrow \quad & wn\log(\sigma^k) + \lambda_2\textstyle\sum_{j=1}^{p}\left|\frac{\beta_j^k}{\sigma^k}\right| \leq C \\
\Longrightarrow \quad & wn\log(\sigma^k) + \lambda_2\left|\frac{\beta_{\tilde{j}}^k}{\sigma^k}\right| \leq C \\
\Longrightarrow \quad & \infty \leq C
\end{aligned}
$$

This is also a contradiction! Thus, (E.5) *also holds.*

Based on Lemma E.2.1, we know that $\beta_j^k = 0, \forall j$, i.e., $\boldsymbol{\beta}^k = \mathbf{0}$, and $y_i^k = \beta_0^k + \boldsymbol{u}_i^k\boldsymbol{\beta}^k, \forall i$. As a result, we have

$$
y_i^k = \beta_0^k, \forall i \tag{E.6}
$$

This contradicts the fact that $y_i^k = t_i$ for $i \in O$, where t_i's are known and not equal. Therefore, $\sigma^k \neq 0$, and thus $\tilde{\sigma}^k = 1/\sigma^k \nrightarrow \infty$. In other words, there exists constants $M_{\tilde{\sigma}}^{\min}$ and $M_{\tilde{\sigma}}^{\max}$ such that $0 < M_{\tilde{\sigma}}^{\min} \leq \tilde{\sigma}^k \leq M_{\tilde{\sigma}}^{\max}$.

E.2.2 $\boldsymbol{U}^k$

From (E.2), we have

$$
\begin{aligned}
& \|\mathcal{P}_\Omega(\boldsymbol{S}-\boldsymbol{U}^k\boldsymbol{V}^k)\|_F^2+\lambda_1(\|\boldsymbol{U}^k\|_F^2+\|\boldsymbol{V}^k\|_F^2)+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)+ \\
& \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{U}^k\|_F^2+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{U}^k\|_F^2+w(-n\log\tilde{\sigma}^k+\|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1)\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{U}^k\|_F^2-wn\log\tilde{\sigma}^k\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{U}^k\|_F^2\leq C+wn\log\tilde{\sigma}^k \\
\Longrightarrow\ & \|\boldsymbol{U}^k\|_F^2\leq(C+wn\log M_{\tilde{\sigma}}^{max})/\lambda_1
\end{aligned}
$$

As a result, let $M_{\boldsymbol{U}}=(C+wn\log M_{\tilde{\sigma}}^{max})/\lambda_1$, we have $\|\boldsymbol{U}^k\|_F^2\leq M_{\boldsymbol{U}}$.

E.2.3 $\boldsymbol{V}^k$

From (E.2), we have

$$
\begin{aligned}
& \|\mathcal{P}_\Omega(\boldsymbol{S}-\boldsymbol{U}^k\boldsymbol{V}^k)\|_F^2+\lambda_1(\|\boldsymbol{U}^k\|_F^2+\|\boldsymbol{V}^k\|_F^2)+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)+ \\
& \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{V}^k\|_F^2+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{V}^k\|_F^2+w(-n\log\tilde{\sigma}^k+\|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}\tilde{\boldsymbol{\beta}}^k\|_1)\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{V}^k\|_F^2-wn\log\tilde{\sigma}^k\leq C \\
\Longrightarrow\ & \lambda_1\|\boldsymbol{V}^k\|_F^2\leq C+wn\log\tilde{\sigma}^k \\
\Longrightarrow\ & \|\boldsymbol{V}^k\|_F^2\leq(C+wn\log M_{\tilde{\sigma}}^{max})/\lambda_1
\end{aligned}
$$

As a result, let $M_{\boldsymbol{V}}=(C+wn\log M_{\tilde{\sigma}}^{max})/\lambda_1$, we have $\|\boldsymbol{V}^k\|_F^2\leq M_{\boldsymbol{V}}$.

E.2.4 $\tilde{\boldsymbol{\beta}}^k$

From (E.2), we have

$$
\begin{aligned}
& \|\mathcal{P}_\Omega(\boldsymbol{S}-\boldsymbol{U}^k\boldsymbol{V}^k)\|_F^2+\lambda_1(\|\boldsymbol{U}^k\|_F^2+\|\boldsymbol{V}^k\|_F^2)+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)+ \\
& \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\le C \\
\Longrightarrow\ & \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)\le C \\
\Longrightarrow\ & \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1+w(-n\log\tilde{\sigma}^k+\|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1)\le C \\
\Longrightarrow\ & \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1-wn\log\tilde{\sigma}^k\le C \\
\Longrightarrow\ & \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\le C+wn\log\tilde{\sigma}^k \\
\Longrightarrow\ & \|\tilde{\boldsymbol{\beta}}^k\|_1\le(C+wn\log M_{\tilde{\sigma}}^{max})/\lambda_2
\end{aligned}
$$

As a result, let $M_{\tilde{\boldsymbol{\beta}}}=(C+wn\log M_{\tilde{\sigma}}^{max})/\lambda_1$, we have $\tilde{\boldsymbol{\beta}}^k\le M_{\tilde{\boldsymbol{\beta}}}$.

E.2.5 $\tilde{\beta}_0^k$

From (E.2), we have

$$
\begin{aligned}
& \|\mathcal{P}_\Omega(\boldsymbol{S}-\boldsymbol{U}^k\boldsymbol{V}^k)\|_F^2+\lambda_1(\|\boldsymbol{U}^k\|_F^2+\|\boldsymbol{V}^k\|_F^2)+w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)+ \\
& \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1\le C \\
\Longrightarrow\ & w\ell(\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k)\le C \\
\Longrightarrow\ & w(-n\log\tilde{\sigma}^k+\|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1)\le C \\
\Longrightarrow\ & \|\tilde{\sigma}^k\boldsymbol{y}^k-\mathbf{1}_n\tilde{\beta}_0^k-\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1\le C/w+n\log\tilde{\sigma}^k \\
\Longrightarrow\ & \textstyle\sum_i|\tilde{\sigma}^k y_i^k-\tilde{\beta}_0^k-\boldsymbol{u}_i^k\tilde{\boldsymbol{\beta}}^k|\le C/w+n\log\tilde{\sigma}^k \\
\Longrightarrow\ & |\tilde{\sigma}^k y_{\tilde{i}}^k-\tilde{\beta}_0^k-\boldsymbol{u}_{\tilde{i}}^k\tilde{\boldsymbol{\beta}}^k|\le C/w+n\log\tilde{\sigma}^k,\ \ \exists\tilde{i},\tilde{i}\in O \\
\Longrightarrow\ & |\tilde{\beta}_0-(\tilde{\sigma}t_{\tilde{i}}-\boldsymbol{u}_{\tilde{i}}\tilde{\boldsymbol{\beta}})|\le C/w+n\log\tilde{\sigma},\ \ \because\text{for any }\tilde{i}\in O,\text{we know that its TTF }\ y_{\tilde{i}}=t_{\tilde{i}} \\
\Longrightarrow\ & |\tilde{\beta}_0^k|-|\tilde{\sigma}^k t_{\tilde{i}}-\boldsymbol{u}_{\tilde{i}}^k\tilde{\boldsymbol{\beta}}^k|\le C/w+n\log\tilde{\sigma}^k \\
\Longrightarrow\ & |\tilde{\beta}_0^k|\le C/w+n\log\tilde{\sigma}^k+|\tilde{\sigma}^k t_{\tilde{i}}-\boldsymbol{u}_{\tilde{i}}^k\tilde{\boldsymbol{\beta}}^k| \\
\Longrightarrow\ & |\tilde{\beta}_0^k|\le C/w+n\log\tilde{\sigma}^k+\tilde{\sigma}^k t_{\tilde{i}}+|\boldsymbol{u}_{\tilde{i}}^k\tilde{\boldsymbol{\beta}}^k| \\
\Longrightarrow\ & |\tilde{\beta}_0^k|<C/w+n\log\tilde{\sigma}^k+\tilde{\sigma}^k t_{\tilde{i}}+\|\boldsymbol{u}_{\tilde{i}}^k\|_1\|\tilde{\boldsymbol{\beta}}^k\|_1
\end{aligned}
$$

Since $\tilde{\sigma}$, $\boldsymbol{U}$ and $\tilde{\boldsymbol{\beta}}$ are bounded, there exists a constant $M_{\tilde{\beta}_0}$ such that $\tilde{\beta}_0^k\le M_{\tilde{\beta}_0}$.

E.2.6 $y_i^k, \forall i \notin O$

$$
\begin{aligned}
& \|\mathcal{P}_\Omega(\boldsymbol{S} - \boldsymbol{U}^k\boldsymbol{V}^k)\|_F^2 + \lambda_1(\|\boldsymbol{U}^k\|_F^2 + \|\boldsymbol{V}^k\|_F^2) + w\ell(\tilde{\sigma}^k\boldsymbol{y}^k - \mathbf{1}_n\tilde{\beta}_0^k - \boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k) + \\
& \lambda_2\|\tilde{\boldsymbol{\beta}}^k\|_1 \leq C \\
\implies & w\ell(\tilde{\sigma}^k\boldsymbol{y}^k - \mathbf{1}_n\tilde{\beta}_0^k - \boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k) \leq C \\
\implies & w(-n\log\tilde{\sigma}^k + \|\tilde{\sigma}^k\boldsymbol{y}^k - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1) \leq C \\
\implies & \|\tilde{\sigma}^k\boldsymbol{y}^k - \mathbf{1}_n\tilde{\beta}_0^k - \boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1 \leq C/w + n\log\tilde{\sigma}^k \\
\implies & \|\tilde{\sigma}^k\boldsymbol{y}^k\|_1 - \|\mathbf{1}_n\tilde{\beta}_0^k + \boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1 \leq C/w + n\log\tilde{\sigma}^k \\
\implies & \|\tilde{\sigma}^k\boldsymbol{y}^k\|_1 \leq C/w + n\log\tilde{\sigma}^k + \|\mathbf{1}_n\tilde{\beta}_0^k + \boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1 \\
\implies & \|\boldsymbol{y}\|_1 \leq (C/w + n\log\tilde{\sigma}^k + n|\tilde{\beta}_0^k| + \|\boldsymbol{U}^k\tilde{\boldsymbol{\beta}}^k\|_1)/\tilde{\sigma}^k
\end{aligned}
$$

Since $y_i = t_i$ for $i \in O$, $y_i > c_i$ for $i \notin O$ and $\tilde{\sigma}$, $\boldsymbol{U}$ and $\tilde{\boldsymbol{\beta}}$ are bounded, we conclude that there exists constants M_{y_i} such that $c_i < y_i^k < M_{y_i}, \forall i \in O$.

E.3 Proof of Lemma 6.3.2

The objective function $\tilde{\mathcal{F}}$ is shown as follows:

$$
\begin{aligned}
& \tilde{\mathcal{F}}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}) \equiv \|\mathcal{P}_\Omega(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{V})\|_F^2 + \lambda_1(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) + \\
& w\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) + \lambda_2\|\tilde{\boldsymbol{\beta}}\|_1,
\end{aligned}
$$

where $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = -n\log\tilde{\sigma} - \sum_{i=1}^n \tilde{\omega}_i + \sum_{i=1}^n \exp(\tilde{\omega}_i)$ for an SEV/Weibull distribution, $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = -n\log\tilde{\sigma} - \sum_{i=1}^n \tilde{\omega}_i + 2\sum_{i=1}^n \log(1 + \exp(\tilde{\omega}_i))$ for a logistics/loglogistics distribution, $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}}) = \frac{n}{2}\log 2\pi - n\log\tilde{\sigma} + \frac{1}{2}\sum_{i=1}^n \tilde{\omega}_i^2$ for a normal/lognormal distribution, and $\tilde{\omega}_i = \tilde{\sigma}y_i - \tilde{\beta}_0 - \boldsymbol{u}_i\tilde{\boldsymbol{\beta}}$.

It is known that the functions $F(\boldsymbol{U}, \boldsymbol{V}) = \|\mathcal{P}_\Omega(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{V})\|_F^2 + \lambda_1(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2)$, $F(x) = x$, $F(x) = \exp(x)$, $F(x) = \log(x)$, $F(x) = \log(1 + \exp(x))$ are real analytic. In addition, the function $F(x) = \|x\|_1$ is semialgebraic. According to [106], the sum of a real analytic function and a semialgebraic function is subanalytic, and thus satisfies the KL property.

E.4 Proof of Lemma 6.3.3

We know that $f(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}) = \|\mathcal{P}_\Omega(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{V})\|_F^2 + w\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_n\tilde{\beta}_0 - \boldsymbol{U}\tilde{\boldsymbol{\beta}})$. From Lemma 6.3.1, we know that $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{y}, \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}, \tilde{\sigma}$ are bounded. It is easy to check that the derivative of the block-partial gradient $\nabla f^k(\boldsymbol{U}^k)$, $\nabla f^k(\boldsymbol{V}^k)$, $\nabla f^k(\tilde{\beta}_0^k)$, $\nabla f^k(\tilde{\boldsymbol{\beta}}^k)$, $\nabla f^k(\tilde{\sigma}^k)$, $\nabla f^k(\boldsymbol{y}^k)$ are bounded, and thus they are Lipschitz continuous in the bounded set. Due to space reasons, we omit proofs here.

E.5 Proof of Theorem 6.3.1

From Proposition 3, we know that $\{\boldsymbol{\theta}^k\}_{k\geq 1}$ has a finite limit point $\boldsymbol{\theta}^*$. Lemma 6.3.2 indicates that the objective function $\tilde{\mathcal{F}}$ satisfies the KL property around $\boldsymbol{\theta}^*$. In addition, Lemma 6.3.3 shows that the block-partial gradient $\nabla f^k(\boldsymbol{U}^k)$, $\nabla f^k(\boldsymbol{V}^k)$, $\nabla f^k(\tilde{\beta}_0^k)$, $\nabla f^k(\tilde{\boldsymbol{\beta}}^k)$, $\nabla f^k(\tilde{\sigma}^k)$, $\nabla f^k(\boldsymbol{y}^k)$ are Lipschitz continuous. According to the Theorem 2.7 in [**Xu2017**], the we complete the proof.

APPENDIX F

SUPPLEMENTARY MATERIALS OF CHAPTER 7

F.1 Proof of Proposition 4

The proof follows the proof of Lemma 1 in [23]. Specifically, the mode-d matricization of tensor $\mathcal{S}_i$ and $\tilde{\mathcal{B}}$ can be expressed as [23]:

$$\boldsymbol{S}_{i(1)} = \boldsymbol{U}_1 \tilde{\boldsymbol{S}}_{i(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)^\top, \tilde{\boldsymbol{B}}_{(1)} = \boldsymbol{U}_1^\top \boldsymbol{B}_{(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right).$$

Then, we have the following:

$$\begin{aligned}
&\langle \mathcal{B}, \mathcal{S}_i \rangle \\
&= \langle \boldsymbol{B}_{(1)}, \boldsymbol{S}_{i(1)} \rangle \\
&= \langle \boldsymbol{B}_{(1)}, \boldsymbol{U}_1 \tilde{\boldsymbol{S}}_{i(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right)^\top \rangle \\
&= \langle \boldsymbol{U}_1^\top \boldsymbol{B}_{(1)} \left(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2\right), \tilde{\boldsymbol{S}}_{i(1)} \rangle \\
&= \langle \tilde{\boldsymbol{B}}_{(1)}, \tilde{\boldsymbol{S}}_{i(1)} \rangle \\
&= \langle \tilde{\mathcal{B}}, \tilde{\mathcal{S}}_i \rangle
\end{aligned}$$

F.2 Optimization Algorithm for Problem (7.3)

The pseudocode of the algorithm is shown in Table F.1 [21]

Input: A set of tensor samples $\{\mathcal{S}_i \in R^{q_1 \times q_2 \times q_3}\}_{i=1}^{n}$

Output: Low-dimensional representations $\left\{\tilde{\mathcal{S}}_i \in R^{p_1 \times p_2 \times p_3}\right\}_{i=1}^{n}$ of the input tensor samples with maximum variation captured

Algorithm:

Step 1 (Preprocessing): Center the input samples as $\left\{\mathcal{X}_i = \mathcal{S}_i - \bar{\mathcal{S}}\right\}_{i=1}^{n}$, where $\bar{\mathcal{S}} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{S}_i$ is the sample mean

Step 2 (Initialization): Calculate the eigen-decomposition of $\boldsymbol{\Phi}_{(d)}^{*} = \sum_{i=1}^{n} \boldsymbol{X}_{i(d)} \boldsymbol{X}_{i(d)}^{\top}$ and set $\boldsymbol{U}_d$ to consist of the eigenvectors corresponding to the most significant p_d eigenvalues, for $d = 1, \ldots 3$

Step 3 (Local optimization):

(i) Calculate $\left\{\tilde{\mathcal{X}}_i = \mathcal{X}_i \times_1 \boldsymbol{U}_1^{\top} \times_2 \boldsymbol{U}_2^{\top} \times_3 \boldsymbol{U}_3^{\top}\right\}_{i=1}^{n}$

(ii) Calculate $\Psi_0 = \sum_{i=1}^{n} ||\tilde{\mathcal{X}}_i||_F^2$ (the mean of $\left\{\tilde{\mathcal{X}}_i\right\}_{i=1}^{n}$ is all zero since $\{\mathcal{X}_i\}_{i=1}^{n}$is centered.

(iii) For $l = 1 : l_{max}$

– For $d = 1 : 3$

Calculate the eigen-decomposition of $\boldsymbol{\Phi}_{(d)}$ and set $\boldsymbol{U}_d$ to consist of the eigenvectors corresponding to the most significant p_d eigenvalues, for $d = 1, \ldots 3$, where $\boldsymbol{\Phi}_{(d)} = \sum_{i=1}^{N} \left(\boldsymbol{S}_{i(d)} - \bar{\boldsymbol{S}}_{(d)}\right) \cdot \boldsymbol{A}_d \cdot \boldsymbol{A}_d^{\top} \cdot \left(\boldsymbol{S}_{i(d)} - \bar{\boldsymbol{S}}_{(d)}\right)^{\top}$ and

$$\boldsymbol{A}_d = \begin{cases} \boldsymbol{U}_2 \otimes \boldsymbol{U}_3, & \text{if } d = 1 \\ \boldsymbol{U}_3 \otimes \boldsymbol{U}_1, & \text{if } d = 2 \\ \boldsymbol{U}_1 \otimes \boldsymbol{U}_2, & \text{if } d = 3 \end{cases},$$

– Calculate $\left\{\tilde{\mathcal{S}}_i\right\}_{i=1}^{n}$ and Ψ_l

– If $\Psi_l - \Psi_{l-1} < \eta$, break and go to Step 4.

Step 4 (Projection): The feature tensor after projection is obtained as $\left\{\tilde{\mathcal{S}}_i = \mathcal{S}_i \times_1 \boldsymbol{U}_1^{\top} \times_2 \boldsymbol{U}_2^{\top} \times_3 \boldsymbol{U}_3^{\top}\right\}_{i=1}^{n}$

Table F.1: Pseudocode implementation of the MPCA algorithm [21].

F.3 Proof of Proposition 5

Without loss of generality, we let $d = 1$. Based on the CP decomposition, tensor $\mathcal{B}$ has the following properties [23]:

$$vec(\mathcal{B}) = (\boldsymbol{B}_3 \odot \boldsymbol{B}_2 \odot \boldsymbol{B}_1)\mathbf{1}_k,$$

$$\boldsymbol{B}_{(1)} = \boldsymbol{B}_1(\boldsymbol{B}_3 \odot \boldsymbol{B}_2)^\top$$

Recall the optimization problem is

$$\arg\max_{\boldsymbol{\theta}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \left\langle (\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2 \odot \tilde{\boldsymbol{B}}_1)\mathbf{1}_k, vec(\tilde{\mathcal{S}}_i) \right\rangle}{\sigma} \right) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\}$$

Given $\tilde{\boldsymbol{B}}_{\neq 1}$,the inner product in the optimization is

$$\begin{aligned}
& \langle (\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2 \odot \tilde{\boldsymbol{B}}_1)\mathbf{1}_k, vec(\tilde{\mathcal{S}}_i) \rangle \\
= \; & \langle vec(\tilde{\mathcal{B}}), vec(\tilde{\mathcal{S}}_i) \rangle \\
= \; & \langle \tilde{\mathcal{B}}, \tilde{\mathcal{S}}_i \rangle \\
= \; & \langle \tilde{\boldsymbol{B}}_{(1)}, \tilde{\boldsymbol{S}}_{i(1)} \rangle \\
= \; & \langle \tilde{\boldsymbol{B}}_1(\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2)^\top, \tilde{\boldsymbol{S}}_{i(1)} \rangle \\
= \; & \langle \tilde{\boldsymbol{B}}_1, \tilde{\boldsymbol{S}}_{i(1)}(\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2) \rangle \\
= \; & \langle \tilde{\boldsymbol{B}}_1, \boldsymbol{X}_{1,i} \rangle
\end{aligned}$$

where $\boldsymbol{X}_{1,i} = \tilde{\boldsymbol{S}}_{i(1)}(\tilde{\boldsymbol{B}}_3 \odot \tilde{\boldsymbol{B}}_2)$. Therefore, the optimization problem can be re-expressed as

$$\arg\max_{\boldsymbol{\theta}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \langle \tilde{\boldsymbol{B}}_1, \boldsymbol{X}_{1,i} \rangle}{\sigma} \right) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_1\right) \right\}$$

F.4 Invariant Property of Optimization Problem (7.9)

Recall optimization problem (7.9)

$$\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{-n\ln\sigma+\sum_{i=1}^{n}\ln f\left(\frac{y_i-\alpha-\left\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right)-r(\frac{\tilde{\boldsymbol{B}}_d}{\sigma})\right\}$$

Consider the transformation $y_i'=by_i,\alpha'=b\alpha,\tilde{\boldsymbol{B}}_d'=b\tilde{\boldsymbol{B}}_d,\sigma'=b\sigma$ where $b>0$, we have

$$\begin{aligned}
&\arg\max_{\tilde{\boldsymbol{B}}_d',\sigma',\alpha'}\left\{-n\ln\sigma'+\textstyle\sum_{i=1}^{n}\ln f\left(\frac{y_i'-\alpha'-\left\langle\tilde{\boldsymbol{B}}_d',\boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right)-r(\frac{\tilde{\boldsymbol{B}}_d'}{\sigma'})\right\}\\
\Longleftrightarrow\quad&\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{-n\ln(b\sigma)+\textstyle\sum_{i=1}^{n}\ln f\left(\frac{by_i-b\alpha-\left\langle b\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{b\sigma}\right)-r(\frac{b\tilde{\boldsymbol{B}}_d}{b\sigma})\right\}\\
\Longleftrightarrow\quad&\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{-n\ln(b\sigma)+\textstyle\sum_{i=1}^{n}\ln f\left(\frac{by_i-b\alpha-b\left\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{b\sigma}\right)-r(\frac{b\tilde{\boldsymbol{B}}_d}{b\sigma})\right\}\\
\Longleftrightarrow\quad&\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{-n\ln b-n\ln\sigma+\textstyle\sum_{i=1}^{n}\ln f\left(\frac{y_i-\alpha-\left\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right)-r(\frac{\tilde{\boldsymbol{B}}_d}{\sigma})\right\}\\
\Longleftrightarrow\quad&\arg\max_{\tilde{\boldsymbol{B}}_d,\sigma,\alpha}\left\{-n\ln\sigma+\textstyle\sum_{i=1}^{n}\ln f\left(\frac{y_i-\alpha-\left\langle\tilde{\boldsymbol{B}}_d,\boldsymbol{X}_{d,i}\right\rangle}{\sigma}\right)-r(\frac{\tilde{\boldsymbol{B}}_d}{\sigma})\right\}
\end{aligned}$$

F.5 Proof of Proposition 6

Based on the Tucker decomposition, tensor $\mathcal{B}$ has the following properties [23]:

$$\boldsymbol{B}_{(1)}=\boldsymbol{B}_1\boldsymbol{G}_{(1)}(\boldsymbol{B}_3\otimes\boldsymbol{B}_2)^\top,$$

$$vec(\mathcal{B})=(\boldsymbol{B}_3\otimes\boldsymbol{B}_2\otimes\boldsymbol{B}_1)vec(\mathcal{G}).$$

Recall the optimization problem is

$$\arg\max_{\boldsymbol{\theta}}\left\{-n\ln\sigma+\sum_{i=1}^{n}\ln f\left(\frac{y_i-\alpha-\left\langle\tilde{\mathcal{G}}\times_1\tilde{\boldsymbol{B}}_1\times_2\tilde{\boldsymbol{B}}_2\times_3\tilde{\boldsymbol{B}}_3,\tilde{\mathcal{S}}_i\right\rangle}{\sigma}\right)-r(\tilde{\mathcal{G}})-\sum_{d=1}^{3}r\left(\tilde{\boldsymbol{B}}_d\right)\right\},$$

Given $\left\{\tilde{\boldsymbol{B}}_1,\tilde{\boldsymbol{B}}_2,\tilde{\boldsymbol{B}}_3\right\}$, the inner product in the optimization can be expressed as

$$\begin{aligned}
&\langle\tilde{\mathcal{G}}\times_1\tilde{\boldsymbol{B}}_1\times_2\tilde{\boldsymbol{B}}_2\times_3\tilde{\boldsymbol{B}}_3,\tilde{\mathcal{S}}_i\rangle\\
=\;&\langle\tilde{\mathcal{B}},\tilde{\mathcal{S}}_i\rangle\\
=\;&\langle vec(\tilde{\mathcal{B}}),vec(\tilde{\mathcal{S}}_i)\rangle\\
=\;&\langle(\tilde{\boldsymbol{B}}_3\otimes\tilde{\boldsymbol{B}}_2\otimes\tilde{\boldsymbol{B}}_1)vec(\tilde{\mathcal{G}}),vec(\tilde{\mathcal{S}}_i)\rangle\\
=\;&\langle vec(\tilde{\mathcal{G}}),(\tilde{\boldsymbol{B}}_3\otimes\tilde{\boldsymbol{B}}_2\otimes\tilde{\boldsymbol{B}}_1)^\top vec(\tilde{\mathcal{S}}_i)\rangle
\end{aligned}$$

Therefore, the optimization problem can be re-expressed as

$$\arg\max_{\tilde{\mathcal{G}}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \left\langle vec(\tilde{\mathcal{G}}), (\tilde{\boldsymbol{B}}_3 \otimes \tilde{\boldsymbol{B}}_2 \otimes \tilde{\boldsymbol{B}}_1)^\top vec(\tilde{\mathcal{S}}_i) \right\rangle}{\sigma} \right) - r(\tilde{\mathcal{G}}) \right\},$$

F.6 Proof for Proposition 7

Without loss of generality, we let $d = 1$. Recall the optimization problem is

$$\arg\max_{\boldsymbol{\theta}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \left\langle \tilde{\mathcal{G}} \times_1 \tilde{\boldsymbol{B}}_1 \times_2 \tilde{\boldsymbol{B}}_2 \times_3 \tilde{\boldsymbol{B}}_3, \tilde{\mathcal{S}}_i \right\rangle}{\sigma} \right) \right.$$
$$\left. -r(\tilde{\mathcal{G}}) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\},$$

Given $\tilde{\mathcal{G}}$ and $\tilde{\boldsymbol{B}}_{\neq 1}$, the inner product in the optimization can be expressed as

$$\begin{aligned}
& \langle \mathcal{G} \times_1 \boldsymbol{B}_1 \times_2 \boldsymbol{B}_2 \times_3 \boldsymbol{B}_3, \tilde{\mathcal{S}}_i \rangle \\
= \; & \langle \tilde{\mathcal{B}}, \tilde{\mathcal{S}}_i \rangle \\
= \; & \langle \tilde{\boldsymbol{B}}_{(1)}, \tilde{\boldsymbol{S}}_{i(1)} \rangle \\
= \; & \langle \boldsymbol{B}_1 \boldsymbol{G}_{(1)} (\boldsymbol{B}_3 \otimes \boldsymbol{B}_2)^\top, \tilde{\boldsymbol{S}}_{i(1)} \rangle \\
= \; & \langle \boldsymbol{B}_1, \tilde{\boldsymbol{S}}_{i(d)} (\boldsymbol{B}_3 \otimes \boldsymbol{B}_2) \boldsymbol{G}_{(1)}^\top \rangle \\
= \; & \langle \boldsymbol{B}_1, \boldsymbol{X}_{1,i} \rangle
\end{aligned}$$

where $\boldsymbol{X}_{1,i} = \tilde{\boldsymbol{S}}_{i(1)} (\boldsymbol{B}_3 \otimes \boldsymbol{B}_2) \boldsymbol{G}_{(1)}^\top$. Therefore, the optimization problem can be re-expressed as

$$\arg\max_{\boldsymbol{\theta}} \left\{ -n \ln \sigma + \sum_{i=1}^{n} \ln f \left(\frac{y_i - \alpha - \langle \boldsymbol{B}_1, \boldsymbol{X}_{1,i} \rangle}{\sigma} \right) - r(\tilde{\mathcal{G}}) - \sum_{d=1}^{3} r\left(\tilde{\boldsymbol{B}}_d\right) \right\}.$$

REFERENCES

[1] N. Z. Gebraeel, M. A. Lawley, R. Li, and J. K. Ryan, "Residual-life distributions from component degradation signals: A bayesian approach," *IIE Transactions*, vol. 37, no. 6, pp. 543–557, 2005.

[2] Z.-S. Ye and N. Chen, "The inverse gaussian process as a degradation model," *Technometrics*, vol. 56, no. 3, pp. 302–311, 2014.

[3] X. Wang and D. Xu, "An inverse gaussian process model for degradation data," *Technometrics*, vol. 52, no. 2, pp. 188–197, 2010.

[4] Z.-S. Ye, N. Chen, and Y. Shen, "A new class of wiener process models for degradation analysis," *Reliability Engineering & System Safety*, vol. 139, pp. 58–67, 2015.

[5] N. Chen, Z.-S. Ye, Y. Xiang, and L. Zhang, "Condition-based maintenance using the inverse gaussian degradation model," *European Journal of Operational Research*, vol. 243, no. 1, pp. 190–199, 2015.

[6] C. Park and W. J. Padgett, "New cumulative damage models for failure using stochastic processes as initial damage," *IEEE Transactions on Reliability*, vol. 54, no. 3, pp. 530–540, 2005.

[7] Y. Shu, Q. Feng, and D. W. Coit, "Life distribution analysis based on lévy subordinators for degradation with random jumps," *Naval Research Logistics (NRL)*, vol. 62, no. 6, pp. 483–492, 2015.

[8] Y. Zhang and H. Liao, "Analysis of destructive degradation tests for a product with random degradation initiation time," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 516–527, 2015.

[9] J. Bogdanoff, F Kozin, and H Saunders, *Probabilistic models of cumulative damage*, 1988.

[10] J. P. Kharoufeh and S. M. Cox, "Stochastic models for degradation-based reliability," *IIE Transactions*, vol. 37, no. 6, pp. 533–542, 2005.

[11] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, IEEE, 2008, pp. 1–9.

[12] X. Fang, K. Paynabar, and N. Gebraeel, "Image-based prognostics using penalized tensor regression," *ArXiv preprint arXiv:1706.03423*, 2017.

[13] N. Gebraeel, "Sensory-updated residual life distributions for components with exponential degradation patterns," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 382–393, 2006.

[14] N. Jin, S. Zhou, and T.-S. Chang, *Identification of impacting factors of surface defects in hot rolling processes using multi-level regression analysis*. Society of Manufacturing Engineers, 2000.

[15] M. Yuan and Y. Lin, "On the non-negative garrotte estimator," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 143–161, 2007.

[16] K. Paynabar, J. Jin, and M. P. Reed, "Informative sensor and feature selection via hierarchical nonnegative garrote," *Technometrics*, vol. 57, no. 4, pp. 514–523, 2015.

[17] J. O. Ramsay, *Functional data analysis*. Wiley Online Library, 2006.

[18] Twitter, "Http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/10/," in, 2015.

[19] X. Fang, R. Zhou, and N. Gebraeel, "An adaptive functional regression-based prognostic model for applications with missing data," *Reliability Engineering & System Safety*, vol. 133, pp. 266–274, 2015.

[20] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[21] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Mpca: Multilinear principal component analysis of tensor objects," *IEEE transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[22] J. Xu, Y. Wang, and L. Xu, "Phm-oriented integrated fusion prognostics for aircraft engines based on sensor data," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1124–1132, 2014.

[23] X. Li, J. Qian, and G.-g. Wang, "Fault prognostic based on hybrid method of state judgment and regression," *Advances in Mechanical Engineering*, vol. 5, p. 149 562, 2013.

[24] R. Moghaddass and M. J. Zuo, "An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process," *Reliability Engineering & System Safety*, vol. 124, pp. 92–104, 2014.

[25] K. Javed, R. Gouriveau, and N. Zerhouni, "Novel failure prognostics approach with dynamic thresholds for machine degradation," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, IEEE, 2013, pp. 4404–4409.

[26] E. Ramasso and R. Gouriveau, "Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions," in *Prognostics and Health Management Conference, 2010. PHM'10.*, IEEE, 2010, pp. 1–10.

[27] M. El-Koujok, R. Gouriveau, and N. Zerhouni, "Reducing arbitrary choices in model building for prognostics: An approach by applying parsimony principle on an evolving neuro-fuzzy system," *Microelectronics reliability*, vol. 51, no. 2, pp. 310–320, 2011.

[28] R. Gouriveau and N. Zerhouni, "Connexionist-systems-based long term prediction approaches for prognostics," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 909–920, 2012.

[29] R. Ishibashi and C. L. N. Júnior, "Gfrbs-phm: A genetic fuzzy rule-based system for phm with improved interpretability," in *Prognostics and Health Management (PHM), 2013 IEEE Conference on*, IEEE, 2013, pp. 1–7.

[30] S. Jianzhong, Z. Hongfu, Y. Haibin, and M. Pecht, "Study of ensemble learning-based fusion prognostics," in *Prognostics and Health Management Conference, 2010. PHM'10.*, IEEE, 2010, pp. 1–7.

[31] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, IEEE, 2008, pp. 1–6.

[32] K. Liu, N. Z. Gebraeel, and J. Shi, "A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 652–664, 2013.

[33] K. Liu and S. Huang, "Integration of data fusion methodology and degradation modeling process to improve prognostics," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 344–354, 2016.

[34] K. Liu, A. Chehade, and C. Song, "Optimize the signal quality of the composite health index via data fusion for degradation modeling and prognostic analy-

sis," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 3, pp. 1504–1514, 2017.

[35] W. Q. Meeker and L. A. Escobar, *Statistical methods for reliability data*. John Wiley & Sons, 2014.

[36] K. Karhunen, *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Universitat Helsinki, 1947, vol. 37.

[37] X. Fang, N. Z. Gebraeel, and K. Paynabar, "Scalable prognostic models for large-scale condition monitoring applications," *IISE Transactions*, vol. 49, no. 7, pp. 698–710, 2017.

[38] L. Breiman, "Better subset regression using the nonnegative garrote," *Technometrics*, vol. 37, no. 4, pp. 373–384, 1995.

[39] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[40] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[41] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[42] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[43] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[44] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[45] J. Fan and I. Gijbels, *Local polynomial modelling and its applications: Monographs on statistics and applied probability 66*. CRC Press, 1996, vol. 66.

[46] C. Loader, *Local regression and likelihood*. Springer Science & Business Media, 2006.

[47] J. A. Rice and B. W. Silverman, "Estimating the mean and covariance structure nonparametrically when the data are curves," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 233–243, 1991.

[48] F. Yao, H.-G. Müller, and J.-L. Wang, "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 577–590, 2005.

[49] H.-G. Müller and Y. Zhang, "Time-varying functional regression for predicting remaining lifetime distributions from longitudinal trajectories," *Biometrics*, vol. 61, no. 4, pp. 1064–1075, 2005.

[50] D. K. Frederick, J. A. DeCastro, and J. S. Litt, "User's guide for the commercial modular aero-propulsion system simulation (c-mapss)," 2007.

[51] P. Ratliff, P. Garbett, and W. Fischer, "The new siemens gas turbine sgt5-8000h for more customer benefit," *VGB powertech*, vol. 87, no. 9, pp. 128–132, 2007.

[52] N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran, "Residual life predictions from vibration-based degradation signals: A neural network approach," *IEEE Transactions on industrial electronics*, vol. 51, no. 3, pp. 694–700, 2004.

[53] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[54] T. Heger and M. C. Pandit, "Optical wear assessment system for grinding tools," *Journal of Electronic Imaging*, vol. 13, no. 3, pp. 450–462, 2004.

[55] D. Simon and D. L. Simon, "Aircraft turbofan engine health estimation using constrained kalman filtering," in *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference*, American Society of Mechanical Engineers, 2003, pp. 485–492.

[56] T. Kobayashi and D. L. Simon, "Hybrid kalman filter approach for aircraft engine in-flight diagnostics: Sensor fault detection case," *Journal of engineering for gas turbines and power*, vol. 129, no. 3, pp. 746–754, 2007.

[57] K. Salahshoor, M. Mosallaei, and M. Bayat, "Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended kalman filter algorithm," *Measurement*, vol. 41, no. 10, pp. 1059–1076, 2008.

[58] K. Goebel and P. Bonissone, "Prognostic information fusion for constant load systems," in *Information Fusion, 2005 8th International Conference on*, IEEE, vol. 2, 2005, 9–pp.

[59] Q. Sun, "Sensor fusion for vehicle health monitoring and degradation detection," in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, IEEE, vol. 2, 2002, pp. 1422–1427.

[60] B. Saha, K. Goebel, and J. Christophersen, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 293–308, 2009.

[61] K. Le Son, M. Fouladirad, A. Barros, E. Levrat, and B. Iung, "Remaining useful life estimation based on stochastic deterioration models: A comparative study," *Reliability Engineering & System Safety*, vol. 112, pp. 165–175, 2013.

[62] E. Ramasso, M. Rombaut, and N. Zerhouni, "Joint prediction of continuous and discrete states in time-series based on belief functions," *IEEE transactions on cybernetics*, vol. 43, no. 1, pp. 37–50, 2013.

[63] R. R. Zhou, N. Serban, and N. Gebraeel, "Degradation modeling applied to residual lifetime prediction using functional data analysis," *The Annals of Applied Statistics*, pp. 1586–1610, 2011.

[64] R. Zhou, N. Gebraeel, and N. Serban, "Degradation modeling and monitoring of truncated degradation signals," *IIE Transactions*, vol. 44, no. 9, pp. 793–803, 2012.

[65] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[66] P. Businger and G. H. Golub, "Linear least squares solutions by householder transformations," *Numerische Mathematik*, vol. 7, no. 3, pp. 269–276, 1965.

[67] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing qr factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.

[68] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[69] L. Doray, "Ibnr reserve under a loglinear location-scale regression model," in *Casualty Actuarial Society Forum Casualty Actuarial Society*, vol. 2, 1994, pp. 607–652.

[70] C. Lu, L. Tao, and H. Fan, "An intelligent approach to machine component health prognostics by utilizing only truncated histories," *Mechanical Systems and Signal Processing*, vol. 42, no. 1-2, pp. 300–313, 2014.

[71] M. Yu and D. Wang, "Model-based health monitoring for a vehicle steering system with multiple faults of unknown types," *IEEE Transactions on industrial electronics*, vol. 61, no. 7, pp. 3574–3586, 2014.

[72] M. Yu, D. Wang, and M. Luo, "Model-based prognosis for hybrid systems with mode-dependent degradation behaviors," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 546–554, 2014.

[73] E. Ramasso and R. Gouriveau, "Remaining useful life estimation by classification of predictions based on a neuro-fuzzy system and theory of belief functions," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 555–566, 2014.

[74] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Estimating remaining useful life with three-source variability in degradation modeling," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 167–190, 2014.

[75] Z.-Q. Wang, C.-H. Hu, W. Wang, and X.-S. Si, "An additive wiener process-based prognostic model for hybrid deteriorating systems," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 208–222, 2014.

[76] K. A. Doksum and A. Hbyland, "Models for variable-stress accelerated life testing experiments based on wener processes and the inverse gaussian distribution," *Technometrics*, vol. 34, no. 1, pp. 74–82, 1992.

[77] M. Li and W. Q. Meeker, "Application of bayesian methods in reliability data analyses," *Journal of Quality Technology*, vol. 46, no. 1, pp. 1–23, 2014.

[78] B. Wu, Z. Tian, and M. Chen, "Condition-based maintenance optimization using neural network-based health condition prediction," *Quality and Reliability Engineering International*, vol. 29, no. 8, pp. 1151–1163, 2013.

[79] W. Peng, H.-Z. Huang, M. Xie, Y. Yang, and Y. Liu, "A bayesian approach for system reliability analysis with multilevel pass-fail, lifetime and degradation data sets," *IEEE Transactions on Reliability*, vol. 62, no. 3, pp. 689–699, 2013.

[80] M. A. Herzog, T. Marwala, and P. S. Heyns, "Machine and component residual life estimation through the application of neural networks," *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 479–489, 2009.

[81] F. Cadini, E. Zio, and D. Avram, "Model-based monte carlo state estimation for condition-based component replacement," *Reliability Engineering & System Safety*, vol. 94, no. 3, pp. 752–758, 2009.

[82] E. Zio and G. Peloni, "Particle filtering prognostic estimation of the remaining useful life of nonlinear components," *Reliability Engineering & System Safety*, vol. 96, no. 3, pp. 403–409, 2011.

[83] E. Myötyri, U. Pulkkinen, and K. Simola, "Application of stochastic filtering for lifetime prediction," *Reliability Engineering & System Safety*, vol. 91, no. 2, pp. 200–208, 2006.

[84] J. Lawless and M. Crowder, "Covariates and random effects in a gamma process model with application to degradation and failure," *Lifetime Data Analysis*, vol. 10, no. 3, pp. 213–227, 2004.

[85] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: Methods and case studies*. Springer, 2007.

[86] F. Yao and H.-G. Müller, "Functional quadratic regression," *Biometrika*, vol. 97, no. 1, pp. 49–64, 2010.

[87] D. Gervini, "Detecting and handling outlying trajectories in irregularly sampled functional datasets," *The Annals of Applied Statistics*, pp. 1758–1775, 2009.

[88] P. Hall, H.-G. Müller, and F. Yao, "Modelling sparse generalized longitudinal observations with latent gaussian processes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 4, pp. 703–723, 2008.

[89] G. M. James, T. J. Hastie, and C. A. Sugar, "Principal component models for sparse functional data," *Biometrika*, vol. 87, no. 3, pp. 587–602, 2000.

[90] D. Şentürk, L. S. Dalrymple, S. M. Mohammed, G. A. Kaysen, and D. V. Nguyen, "Modeling time-varying effects with generalized and unsynchronized longitudinal data," *Statistics in medicine*, vol. 32, no. 17, pp. 2971–2987, 2013.

[91] D. Şentürk and D. V. Nguyen, "Varying coefficient models for sparse noise-contaminated longitudinal data," *Statistica Sinica*, vol. 21, no. 4, p. 1831, 2011.

[92] Y. Wu and Y. Liu, "Functional robust support vector machines for sparse and irregular longitudinal data," *Journal of computational and Graphical Statistics*, vol. 22, no. 2, pp. 379–395, 2013.

[93] R. R. Zhou, N. Serban, N. Gebraeel, and H.-G. Müller, "A functional time warping approach to modeling and monitoring truncated degradation signals," *Technometrics*, vol. 56, no. 1, pp. 67–77, 2014.

[94] B. J Mercer, "Xvi. functions of positive and negative type, and their connection the theory of integral equations," *Phil. Trans. R. Soc. Lond. A*, vol. 209, no. 441-458, pp. 415–446, 1909.

[95] D. A. Virkler, B. Hillberry, and P. Goel, "The statistical nature of fatigue crack propagation," *Journal of Engineering Materials and Technology*, vol. 101, no. 2, pp. 148–153, 1979.

[96] N. Gebraeel, A. Elwany, and J. Pan, "Residual life predictions in the absence of prior degradation knowledge," *IEEE Transactions on Reliability*, vol. 58, no. 1, pp. 106–117, 2009.

[97] X. Fang, K. Paynabar, and N. Gebraeel, "Multistream sensor fusion-based prognostics model for systems with single failure modes," *Reliability Engineering & System Safety*, vol. 159, pp. 322–331, 2017.

[98] H. Liao and J. Sun, "Nonparametric and semi-parametric sensor recovery in multichannel condition monitoring systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 4, pp. 744–753, 2011.

[99] J. Sun, H. Liao, and B. R. Upadhyaya, "A robust functional-data-analysis method for data recovery in multichannel sensor systems," *IEEE transactions on cybernetics*, vol. 44, no. 8, pp. 1420–1431, 2014.

[100] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd international conference on Machine learning*, ACM, 2005, pp. 713–719.

[101] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[102] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of optimization*, vol. 6, no. 615-640, p. 15, 2010.

[103] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[104] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, no. 1, pp. 102–114, 2017.

[105] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[106] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[107] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[108] S Bagavathiappan, B. Lahiri, T Saravanan, J. Philip, and T Jayakumar, "Infrared thermography for condition monitoring–a review," *Infrared Physics & Technology*, vol. 60, pp. 35–55, 2013.

[109] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Review of vision-based steel surface inspection systems," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 50, 2014.

[110] C. Meola, "Infrared thermography of masonry structures," *Infrared physics & technology*, vol. 49, no. 3, pp. 228–233, 2007.

[111] J. J. Seo, H. Yoon, H. Ha, D. P. Hong, and W. Kim, "Infrared thermographic diagnosis mechnism for fault detection of ball bearing under dynamic loading conditions," in *Advanced materials research*, Trans Tech Publ, vol. 295, 2011, pp. 1544–1547.

[112] M. Pastor, X Balandraud, M Grédiac, and J. Robert, "Applying infrared thermography to study the heating of 2024-t3 aluminium specimens under fatigue loading," *Infrared Physics & Technology*, vol. 51, no. 6, pp. 505–515, 2008.

[113] M Vellvehi, X Perpiñà, G. Lauro, F Perillo, and X Jordà, "Irradiance-based emissivity correction in infrared thermography for electronic applications," *Review of scientific instruments*, vol. 82, no. 11, p. 114 901, 2011.

[114] H. Yan, K. Paynabar, and J. Shi, "Image-based process monitoring using low-rank tensor decomposition," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 216–227, 2015.

[115] X. Liu, K. Yeo, and J. Kalagnanam, "Statistical modeling for spatio-temporal degradation data," *ArXiv preprint arXiv:1609.07217*, 2016.

[116] N. Cressie and C. K. Wikle, *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.

[117] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[118] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-youngi decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[119] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[120] J. De Leeuw, "Block-relaxation algorithms in statistics," in *Information systems and data analysis*, Springer, 1994, pp. 308–324.

[121] K. Lange, *Numerical analysis for statisticians*. Springer Science & Business Media, 2010.

[122] N. Städler, P. Bühlmann, and S. Van De Geer, " 1-penalization for mixture regression models," *Test*, vol. 19, no. 2, pp. 209–256, 2010.

[123] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[124] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, *et al.*, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[125] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[126] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 540–552, 2013.

[127] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[128] X. Fang, K. Paynabar, and N. Gebraeel, "Real-time predictive analytics using degradation image data," in *Reliability and Maintainability Symposium (RAMS), 2018 Annual*, IEEE, 2018.

[129] F. Yao, H.-G. Müller, A. J. Clifford, S. R. Dueker, J. Follett, Y. Lin, B. A. Buchholz, and J. S. Vogel, "Shrinkage estimation for functional principal component scores with application to the population kinetics of plasma folate," *Biometrics*, vol. 59, no. 3, pp. 676–685, 2003.

VITA

Xiaolei Fang's research interests lie in the field of industrial predictive analytics for High-Dimensional and Big Data applications in the energy, manufacturing, and service sectors. Specifically, he focuses on addressing analytical, computational, and scalability challenges associated with the development of statistical and optimization methodologies for analyzing massive amounts of complex data structures for real-time asset management and optimization. He received his B.S. degree in *Mechanical Engineering* from the University of Science and Technology Beijing, China, in 2008 and an M.S. degree in *Statistics* from the Georgia Institute of Technology, Atlanta, in 2016. He joined the H. Milton Stewart School of Industrial and Systems Engineering (ISyE) at Georgia Institute of Technology as a doctoral student in *Industrial Engineering* in 2014. He was the winner of *the Alice and John Jarvis Ph.D. Student Research Award* (awarded to one Ph.D. student in ISyE per year for outstanding research achievements) and *the INFORMS SAS Data Mining Best Paper Award*. One of his research papers was featured in *Industrial and Systems Engineer (ISE) Magazine*.